HTTP
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2020

M. Nottingham
Fastly

March 1, 2020

# The Cache-Status HTTP Response Header Field
## draft-ietf-httpbis-cache-header-03

## 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 September 2, 2020.

Table of Contents



## 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.





### 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 "%s" extension for case sensitivity defined in [RFC7405].

## 2. The Cache-Status HTTP Response Header Field

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 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 client (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.

Each list member identifies the cache that inserted that 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 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.

This specification defines these parameters:





```
hit          = sh-boolean
fwd          = sh-token
fwd-status   = sh-integer
ttl          = sh-integer
stored       = sh-boolean
collapsed    = sh-boolean
key          = sh-string
```

## 2.1. The hit parameter

"hit", when true, indicates that the request was satisfied by the cache; i.e., it did not go forward, and the response was obtained from the cache (possibly with modifications; e.g., if the request was conditional, a 304 Not Modified could be generated from cache).

"hit" and "fwd" are exclusive; only one of them should appear on each list member.

## 2.2. The fwd parameter

"fwd" indicates why the request went forward.

It can have one of the following values:

- uri-miss - The cache did not contain any responses that matched the request URI

- vary-miss - The cache contained a response that matched the request URI, but could not select a response based upon this

request's headers and stored Vary headers.

- 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)

- stale - The cache was able to select a response for the request, but it was stale

- request - 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

- bypass - The cache was configured to not handle this request





## 2.3. The fwd-status parameter

"fwd-status" indicates what status code the next hop server returned in response to the request. Only meaningful when "fwd" is present; if "fwd-status" is not present but "fwd" is, it defaults to the status code sent in the response.

This parameter is useful to distinguish cases when the next hop server sends a 304 Not Modified response to a conditional request, or a 206 Partial Response due to a range request.

## 2.4. The ttl parameter

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

## 2.5. The stored parameter

"stored" indicates whether the cache stored the forward response; a

true value indicates that it did. Only meaningful when fwd is present.

### 2.6. The collapsed parameter

"collapsed" 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. Only meaningful when fwd is present.

### 2.7. 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; hit
```

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

```
Cache-Status: ExampleCache; hit; ttl=376
```





A stale hit just has negative freshness:

```
Cache-Status: ExampleCache; hit; ttl=-412
```

Whereas a complete miss is:

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

A miss that successfully validated on the back-end server:

```
Cache-Status: ExampleCache; fwd=stale; fwd-status=304
```

A miss that was collapsed with another request:

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

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

```
Cache-Status: ExampleCache; fwd=uri-miss; collapsed=?0
```

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

```
Cache-Status: OriginCache; hit; ttl=1100; collapsed,
              "CDN Company Here"; hit; ttl=545
```

## 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-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/