

Workgroup: Network Working Group
Internet-Draft:
draft-nottingham-cache-trailers-00
Published: 7 March 2021
Intended Status: Standards Track
Expires: 8 September 2021
Authors: M. Nottingham J. Snell

Updating HTTP Caching Policy in Trailers

Abstract

This specification defines how to update caching policy for a response in HTTP trailer fields, after the content has been sent.

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at <https://github.com/mnot/I-D/labels/cache-trailers>.

The most recent (often, unpublished) draft is at <https://mnot.github.io/I-D/cache-trailers/>.

Recent changes are listed at <https://github.com/mnot/I-D/commits/gh-pages/cache-trailers>.

See also the draft's current status in the IETF datatracker, at <https://datatracker.ietf.org/doc/draft-nottingham-cache-trailers/>.

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 8 September 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 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](#)
 - [1.1. Notational Conventions](#)
- [2. The "trailer-update" HTTP Cache Directive](#)
 - [2.1. Examples](#)
- [3. IANA Considerations](#)
- [4. Security Considerations](#)
- [5. Normative References](#)
- [Authors' Addresses](#)

1. Introduction

Web content that is "dynamically" generated -- i.e., with the response body streamed by the server to the client as it is created -- is often assumed to be uncacheable. In practice, though, there are some scenarios where caching is beneficial; for example, when a private cache might be able to reuse a personalised, dynamic response for a period, or when such a response can be shared by a number of clients.

A server choosing a caching policy for such a response faces a conundrum: if an error or other unforeseen condition happens during the generation of the response, that caching policy might be too liberal. Currently, the only available solutions are to:

1. prevent or severely curtail downstream caching, or
2. buffer the response until a caching policy can be confidently assigned.

In both cases, performance suffers; in the former, caching efficiency is less than it could be in the common case, In the latter, the server consumes additional resources and delays the response.

This specification provides a third solution: updating the caching policy in HTTP trailer fields, after the content has been sent. Doing so allows content to be streamed, while caching policy can be determined after the content is actually generated.

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.

2. The "trailer-update" HTTP Cache Directive

The "trailer-update" cache response directive indicates that the caching policy for that response (as indicated by the header field that contains the directive) might be updated by a corresponding trailer field.

When it is present as a cache directive in a header field and a trailer field with the same field name is received, a cache that implements this specification MUST completely replace the stored header field value for that response with the trailer field's value, MUST update its handling of that response to account for the new field value (after any outstanding requests are satisfied), and MUST use that value for the header field in responses to future requests satisfied from that cache entry (i.e., the trailer field is "promoted" to a header field).

In responses where the trailer field value has replaced the header field value, the "trailer-update" directive will have been removed as part of that process. Note that the presence of "trailer-update" does not guarantee that a trailer field will follow.

Caches MAY temporarily store a response that has a caching policy with both the "no-store" and "trailer-update" directives, but MUST NOT reuse that response until the caching policy is updated in a manner that allows it. If the caching policy is not updated or the "no-store" directive is still present in the updated response, the cache MUST immediately and permanently discard the temporarily stored response.

For purposes of calculating a stored response's age ([[I-D.ietf-httpbis-cache](#)], [Section 4.2.3](#)), caches receiving such a trailer SHOULD consider the response_time to be when the trailer is received, but only when calculating resident_time (not response_delay, as that would be counterproductive for the purpose of estimating network delay).

2.1. Examples

Given a resource that supports this specification but encounters no errors in the generation of a response's content, that response might look like this:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: max-age=3600, trailer-update

[body]
```

Caches that do not implement this specification will cache it by the header policy; caches that do implement will see no updates in the trailer and do the same.

If a change in caching policy is encountered during the generation of the response content, the resource can prevent reuse by caches that implement this specification by sending:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: max-age=3600, trailer-update

[body]
Cache-Control: no-store
```

In this case, caches that do not implement this specification will again act as instructed by the header policy, but caches that do implement will see the update in the trailers and prevent reuse of the response after the trailer is received (although it might have been used to satisfy requests that were received in the meantime).

If a resource wishes to prevent non-implementing caches from storing the response, they can send:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: no-store; trailer-update

[body]
Cache-Control: max-age=3600
```

Here, a non-implementing cache will only see "no-store", and so will not store the response. An implementing cache can optimistically store the response based upon "trailer-update", but only allow its reuse after the caching policy is updated to something which permits that in trailers.

Note that when a downstream cache does not implement this specification, and also does not forward a message's trailer section (as allowed by HTTP), any updates will effectively be lost, even if further downstream caches do implement.

3. IANA Considerations

TBD

4. Security Considerations

TBD

5. Normative References

[I-D.ietf-httpbis-cache]

Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP Caching", Work in Progress, Internet-Draft, draft-ietf-httpbis-cache-14, 12 January 2021, <<https://www.ietf.org/archive/id/draft-ietf-httpbis-cache-14.txt>>.

[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>>.

[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>>.

Authors' Addresses

Mark Nottingham
Pahran VIC
Australia

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

James Snell

Email: jasnell@gmail.com