Internet Engineering Task Force (IETF)
Request for Comments: 9213
Category: Standards Track
ISSN: 2070-1721

S. Ludin
Akamai
M. Nottingham
Fastly
Y. Wu
Cloudflare

June 2022

Targeted HTTP Cache Control

draft-ietf-httpbis-targeted-cache-control-04

Abstract

This specification defines a convention for HTTP response header fields that allow cache directives to be targeted at specific caches or classes of caches. It also defines one such header field, the CDN-Cache-Control response header field, which is targeted at content delivery network (CDN) caches.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841[1].

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9213[2].

Copyright Notice

[1] https://www.rfc-editor.org/rfc/rfc7841.html#section-2
[2] https://www.rfc-editor.org/info/rfc9213
[3] https://trustee.ietf.org/license-info

as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.]]], 2nd string ends in: [[[ection 2 of RFC 78 4111 https://www.rfc-editor.org/rfc/rfc7841.html#section-2.Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc921322 https://www.rfc-editor.org/info/rfc9213.Copyright NoticeCopyright (c) 2022 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-info33 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.]]] (at line 6)

Table of Contents

1. Introduction

Modern deployments of HTTP often use multiple layers of caching. For example, a website might use a cache on the origin server itself; it might deploy a caching layer in the same network as the origin server, it might use one or more CDNs that are distributed throughout the Internet, and it might benefit from browser caching as well.

Because it is often desirable to control these different classes of caches separately, some means of targeting cache directives at them is necessary. For example, if a publisher has a mechanism to invalidate the contents of a cache that it has a relationship with (such as a CDN cache), they might be more comfortable assigning a more generous caching policy to it while still wanting to restrict the behavior of other caches.

The HTTP Cache-Control response header field (defined in Section 5.2 of [HTTP-CACHING]) is widely used to direct caching behavior. However, it is relatively undifferentiated; while some cache directives (e.g., s-maxage) are targeted at a specific class of caches (for s-maxage, shared caches), targeting is not consistently available across all existing cache directives (e.g., stale-while-revalidate). This is problematic especially as the number of caching extensions grows along with the number of potential targets.

Some implementations have defined ad hoc control mechanisms to overcome this issue, but their interoperability is low. Section 2 defines a standard framework for targeted cache control using HTTP response headers, and Section 3 defines one such header: the CDN-Cache-Control response header field.

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 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Targeted Cache-Control Header Fields

A Targeted Cache-Control Header Field (hereafter "targeted field") is an HTTP response header field that has the same semantics as the Cache-Control response header field ([HTTP-CACHING], Section 5.2). However, it has a distinct field name that indicates the target for its cache directives.

For example:

```
CDN-Cache-Control: max-age=60
```

is a targeted field that applies to CDNs, as defined in Section 3.

2.1. Syntax

Targeted fields are Dictionary Structured Fields (Section 3.2 of [STRUCTURED-FIELDS]). Each member of the Dictionary is an HTTP cache response directive (Section 5.2.2 of [HTTP-CACHING]) including extension response directives (as per Section 5.2.3 of [HTTP-CACHING]). Note that while targeted fields often have the same syntax as Cache-Control fields, differences in error handling mean that using a Cache-Control parser rather than a Structured Fields parser can introduce interoperability issues.

Because cache directives are not defined in terms of structured data types, it is necessary to map their values into the appropriate types. Section 5.2 of [HTTP-CACHING] defines cache directive values to be either absent, a quoted-string, or a token.

This means that cache directives that have no value will be mapped to a Boolean (Section 3.3.6 of [STRUCTURED-FIELDS]). When the value is a quoted-string, it will be mapped to a String (Section 3.3.3 of [STRUCTURED-FIELDS]), and when it is a token, it will map to a Token (Section 3.3.4 of [STRUCTURED-FIELDS]), an Integer (Section 3.3.1 of [STRUCTURED-FIELDS]), or a Decimal (Section 3.3.2 of [STRUCTURED-FIELDS]), depending on the content of the value.

For example, the max-age directive (Section 5.2.2.1 of [HTTP-CACHING]) has an integer value; no-store (Section 5.2.2.5 of [HTTP-CACHING]) always has a Boolean true value, and no-cache (Section 5.2.2.4 of [HTTP-CACHING]) has a value that can be either Boolean true or a string containing a comma-delimited list of field names.

Implementations MUST NOT generate values that violate these inferred constraints on the cache directive's value. In particular, string values whose first character is not alphabetic or "*" MUST be generated as Strings so that they are not mistaken for other types.

Implementations SHOULD NOT consume values that violate these inferred constraints. For example, a consuming implementation that coerces a max-age with a decimal value into an integer would behave differently than other implementations, potentially causing interoperability issues.

Parameters received on cache directives are to be ignored, unless other handling is explicitly specified.

If a targeted field in a given response is empty, or a parsing error is encountered, that field MUST be ignored by the cache (i.e., it behaves as if the field were not present, likely falling back to other cache-control mechanisms present).

2.2. Cache Behavior

A cache that implements this specification maintains a target list. A target list is an ordered list of the targeted field names that it uses for caching policy, with the order reflecting priority from most applicable to least. The target list might be fixed, user configurable, or generated per request, depending upon the implementation.

For example, a CDN cache might support both CDN-Cache-Control and a header specific to that CDN, ExampleCDN-Cache-Control, with the latter overriding the former. Its target list would be:

```
[ExampleCDN-Cache-Control, CDN-Cache-Control]
```

When a cache that implements this specification receives a response with one or more of the header field names on its target list, the cache MUST select the first (in target-list order) field with a valid, non-empty value and use its value to determine the caching policy for the response, and it MUST ignore the Cache-Control and Expires header fields in that response, unless no valid, non-empty value is available from the listed header fields.

Note that this occurs on a response-by-response basis; if no member of the cache's target list is present, valid, and non-empty, a cache falls back to other cache control mechanisms as required by HTTP [HTTP-CACHING].

Targeted fields that are not on a cache's target list MUST NOT change that cache's behavior and MUST be passed through.

Caches that use a targeted field MUST implement the semantics of the following cache directives:

- max-age
- must-revalidate
- no-store
- no-cache
- private

Furthermore, they SHOULD implement other cache directives (including extension cache directives) that they support in the Cache-Control response header field.

The semantics and precedence of cache directives in a targeted field are the same as those in Cache-Control. In particular, no-store and no-cache make max-age inoperative, and unrecognized extension directives are ignored.

2.3. Interaction with HTTP Freshness

HTTP caching has a single, end-to-end freshness model defined in Section 4.2 of [HTTP-CACHING]. When additional freshness mechanisms are only available to some caches along a request path (for example, using targeted fields), their interactions need to be carefully considered. In particular, a targeted cache might have longer freshness lifetimes available to it than other caches, causing it to serve responses that appear to be prematurely (or even immediately) stale to those other caches, negatively impacting cache efficiency.

For example, a response stored by a CDN cache might be served with the following headers:

```
Age: 1800
Cache-Control: max-age=600
CDN-Cache-Control: max-age=3600
```

From the CDN's perspective, this response is still fresh after being cached for 30 minutes, while from the standpoint of other caches, this response is already stale. See [AGE-PENALTY] for more discussion.

When the targeted cache has a strong coherence mechanism (e.g., the origin server has the ability to proactively invalidate cached responses), it is often desirable to mitigate these effects. Some techniques seen in deployments include:

- Removing the Age header field
- Updating the Date header field value to the current time
- Updating the Expires header field value to the current time, plus any Cache-Control: max-age value

This specification does not place any specific requirements on implementations to mitigate these effects, but definitions of targeted fields can do so.

2.4. Defining Targeted Fields

A targeted field for a particular class of cache can be defined by requesting registration in the "Hypertext Transfer Protocol (HTTP) Field Name Registry" (<https://www.iana.org/assignments/http-fields/>).

Registration requests can use this document as the specification document; in which case, the Comments field should clearly define the class of caches that the targeted field applies to. Alternatively, if other documentation for the field has been created, it can be used as the specification document.

By convention, targeted fields have the suffix "-Cache-Control", e.g., "ExampleCDN-Cache-Control". However, this suffix MUST NOT be used on its own to identify targeted fields; it is only a convention.

3. The CDN-Cache-Control Targeted Field

The CDN-Cache-Control response header field is a targeted field (Section 2) that allows origin servers to control the behavior of CDN caches interposed between them and clients separately from other caches that might handle the response.

It applies to caches that are part of a distributed network that operate on behalf of an origin server (commonly called a CDN).

CDN caches that use CDN-Cache-Control will typically forward this header so that downstream CDN caches can use it as well. However, they MAY remove it when this is undesirable (for example, when configured to do so because it is known not to be used downstream).

3.1. Examples

For example, the following header fields would instruct a CDN cache (i.e., a cache with a target list of `[CDN-Cache-Control]`) to consider the response fresh for 600 seconds, other shared caches to consider the response fresh for 120 seconds, and any remaining caches to consider the response fresh for 60 seconds:

```
Cache-Control: max-age=60, s-maxage=120
CDN-Cache-Control: max-age=600
```

These header fields would instruct a CDN cache to consider the response fresh for 600 seconds, while all other caches would be prevented from storing it:

```
CDN-Cache-Control: max-age=600
Cache-Control: no-store
```

Because CDN-Cache-Control is not present, this header field would prevent all caches from storing the response:

```
Cache-Control: no-store
```

Whereas these would prevent all caches except for CDN caches from storing the response:

```
Cache-Control: no-store
CDN-Cache-Control: none
```

(Note that 'none' is not a registered cache directive; it is here to avoid sending a header field with an empty value, which would be ignored.)

4. IANA Considerations

IANA has registered the following entry in the "Hypertext Transfer Protocol (HTTP) Field Name Registry" defined by [HTTP]:

Field Name:	CDN-Cache-Control
Status:	permanent
Specification Document:	RFC 9213
Comments:	Cache directives targeted at content delivery networks

5. Security Considerations

The security considerations of HTTP caching [HTTP-CACHING] apply.

The ability to carry multiple caching policies on a response can result in confusion about how a response will be cached in different systems, potentially resulting in unintentional reuse of responses with sensitive information. For this reason, care must be exercised.

6. References

6.1. Normative References

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[HTTP-CACHING] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <https://www.rfc-editor.org/info/rfc9111>.

[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>.

[STRUCTURED-FIELDS] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/info/rfc8941>.

6.2. Informative References

[AGE-PENALTY] Cohen, E. and H. Kaplan, "The age penalty and its effect on cache performance", March 2001, <https://dl.acm.org/doi/10.5555/1251440.1251447>.

Authors' Addresses

Stephen Ludin
Akamai
EMail: sludin@ludin.org

Mark Nottingham
Fastly
Prahran
Australia
EMail: mnot@mnot.net
URI: https://www.mnot.net/

Yuchen Wu
Cloudflare
EMail: me@yuchenwu.net