

Workgroup: HTTP

Internet-Draft: draft-cdn-control-header-00

Published: 23 November 2020

Intended Status: Informational

Expires: 27 May 2021

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

The CDN-Cache-Control HTTP Response Header Field

Abstract

This specification defines an HTTP header field that conveys HTTP cache directives to CDN caches.

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at <https://github.com/cdn-specs/control-header/issues>.

The most recent (often, unpublished) draft is at <https://cdn-specs.github.io/control-header/>.

Recent changes are listed at <https://github.com/cdn-specs/control-header/commits/main>.

See also the draft's current status in the IETF datatracker, at <https://datatracker.ietf.org/doc/draft-cdn-control-header/>.

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 27 May 2021.

Copyright Notice

Copyright (c) 2020 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 CDN-Cache-Control Response Header Field](#)
 - [2.1. Examples](#)
 - [2.2. Parsing](#)
- [3. Security Considerations](#)
- [4. Normative References](#)
- [Appendix A. Frequently Asked Questions](#)
 - [A.1. Why not Surrogate-Control?](#)
 - [A.2. Why not mix with Cache-Control?](#)
 - [A.3. Is this just for CDNs?](#)
 - [A.4. What if I use more than one CDN?](#)
- [Authors' Addresses](#)

1. Introduction

Many HTTP origin servers use Content Delivery Networks (i.e., distributed HTTP gateways, usually implementing caches) to speed up distributing their content.

While HTTP defines Cache-Control as a means of controlling cache behaviour for both private caches and shared caches, it is often desirable to give CDN caches separate instructions. To meet this need, this specification defines a separate header field that conveys HTTP cache directives to CDN caches only.

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 CDN-Cache-Control Response Header Field

The CDN-Cache-Control response header field allows origin servers to control the behaviour of CDN caches interposed between them and clients, separately from other caches that might handle the response.

It is a Dictionary Structured Header [[I-D.ietf-httpbis-header-structure](#)], whose members can be any directive registered in the HTTP Cache Directive Registry <https://www.iana.org/assignments/http-cache-directives/http-cache-directives.xhtml>.

When a valid CDN-Cache-Control header field is present in a response, CDN caches MUST ignore the Cache-Control and Expires response headers in that response. As such, CDN-Cache-Control is a wholly separate way to control the CDN cache. Note that this is on a response-by-response basis; if CDN-Cache-Control is not present, CDN caches MAY fall back to other control mechanisms as required by HTTP [[I-D.ietf-httpbis-cache](#)].

The semantics and precedence of cache directives in CDN-Cache-Control are the same as those in Cache-Control. In particular, no-store and no-cache make max-age inoperative.

Caches that use CDN-Cache-Control MUST implement the semantics of the following directives:

- *max-age

- *must-revalidate

- *no-store

- *no-cache

- *private

CDN caches that use CDN-Cache-Control MAY forward this header so that downstream CDN caches can use it as well. However, doing so exposes its value to all downstream clients, which might be undesirable. As a result, CDN caches that process this header field MAY remove it (for example, when configured to do so because it is known not to be used downstream).

A CDN cache that does not use CDN-Cache-Control MUST pass the CDN-Cache-Control header through.

Private caches SHOULD ignore the CDN-Cache-Control header field.

2.1. Examples

For example, the following header fields would instruct a CDN cache to consider the response fresh for 600 seconds, other shared caches for 120 seconds and any remaining caches 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:

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

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, because such a header might not be preserved in all cases)

2.2. Parsing

CDN-Cache-Control is specified as a Structured Field [[I-D.ietf-httpbis-header-structure](#)], and implementations are encouraged to use a parser for that format in the interests of robustness, interoperability and security.

When an implementation parses CDN-Cache-Control as a Structured Field, each directive will be assigned a value. For example, max-age has an integer value; no-store's value is boolean true, and no-cache's value can either be boolean true or a list of field names. Implementations SHOULD NOT accept other values (e.g. coerce a max-age with a decimal value into an integer). Likewise, implementations SHOULD ignore parameters on directives, unless otherwise specified.

However, implementers MAY initially reuse a Cache-Control parser for simplicity. If they do so, they SHOULD observe the following points,

to aid in a smooth transition to a full Structured Field parser and prevent interoperability issues:

*If a directive is repeated in the field value (e.g., "max-age=30, max-age=60"), the last value 'wins' (60, in this case)

*Members of the directives can have parameters (e.g., "max-age=30;a=b;c=d"), which should be ignored unless specified.

3. Security Considerations

The security considerations of HTTP caching [[I-D.ietf-httpbis-cache](#)] apply.

4. Normative References

[I-D.ietf-httpbis-cache]

Fielding, R., Nottingham, M., and J. Reschke, "HTTP Caching", Work in Progress, Internet-Draft, draft-ietf-httpbis-cache-12, 2 October 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-httpbis-cache-12.txt>>.

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

Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", Work in Progress, Internet-Draft, draft-ietf-httpbis-header-structure-19, 3 June 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-httpbis-header-structure-19.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>>.

Appendix A. Frequently Asked Questions

A.1. Why not Surrogate-Control?

The Surrogate-Control header field is used by a variety of cache implementations, but their interpretation of it is not consistent; some only support 'no-store', others support a few directives, and still more support a larger variety of implementation-specific directives. These implementations also differ in how they relate Surrogate-Control to Cache-Control and other mechanisms.

Rather than attempting to align all of these different but well established behaviours (which would likely fail, because many existing deployments depend upon them) or defining a very small subset, a new header field seems more likely to provide clear interoperability without compromising functionality.

A.2. Why not mix with Cache-Control?

An alternative design would be to have CDN caches combine the directives found in Cache-Control and CDN-Cache-Control, considering their union as the directives that must be followed.

While this would be slightly less verbose in some cases, it would make interoperability considerably more complex to achieve. Consider the case when there are syntax errors in the argument of a directive; e.g., 'max-age=n60'. Should that directive be ignored, or does it invalidate the entire header field value? If the directive is ignored in CDN-Cache-Control, should the cache fall back to a value in Cache-Control? And so on.

Also, this approach would make it difficult to direct the CDN cache to store something while directing other caches to avoid storing it (because no-store overrides max-age).

A.3. Is this just for CDNs?

By default, yes. There is often a need to differentiate between CDNs and gateway caches deployed local to the origin server; CDN-Cache-Control allows that.

In some cases, a site might create a CDN by deploying gateway caches and routing traffic to them; this is, after all, how a CDN works at a high level. To support this scenario, gateway caches MAY be configured to process the CDN-Cache-Control header field, but they MUST NOT default to supporting it.

A.4. What if I use more than one CDN?

Individual CDNs can choose to define their own control mechanisms that take precedence over this header field. It is RECOMMENDED that they use a header whose value has the same syntax and semantics, and use a field name in the pattern "CDN_NAME-CDN-Cache-Control"; for example, the Foo CDN might register "Foo-CDN-Cache-Control". When present on a response received by Foo CDN, that header field would override both CDN-Cache-Control and Cache-Control. As with CDN-Cache-Control, Foo CDN could decide whether or not to strip that header from responses before sending them.

Authors' Addresses

Stephen Ludin
Akamai

Email: sludin@ludin.org

Mark Nottingham
Fastly
made in
Pahran VIC
Australia

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

Yuchen Wu
Cloudflare

Email: me@yuchenwu.net