Network Working Group
Internet-Draft
Expires: March 23, 2001

M. Nottingham
Akamai Technologies

September 22, 2000

HTTP Cache Control Extensions for Direct Cache Manipulation
draft-nottingham-cache-extensions-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 23, 2001.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

HTTP/1.1 provides for extensions to Cache-Control headers, which provide new methods of controlling caches. This document specifies extensions which allow content providers more precise control over shared caches.

1. Introduction

Shared caches are sometimes deployed into the Internet with the participation of the providers of content which will be flowing through them. For example, a content provider may use a surrogate HTTP server to improve performance. Likewise, an access provider who controls both the origin servers and caching proxies in their network may wish to optimize the relationship between them.

Although HTTP/1.1 provides mechanisms for controlling caches, it does not provide an efficient means of controlling objects already in cache. In the situations outlined above, it would be useful to allow content providers to manipulate cached objects.

This document provides simple mechanisms to do so, by use of Cache-Control request header extensions.

It should be noted that because of their nature, the extensions lose efficiency when used with a large number of shared caches.

1.1 Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119[2].

An implementation is not compliant if it fails to satisfy one or more of the MUST or REQUIRED level requirements. An implementation that satisfies all the MUST or REQUIRED level and all the SHOULD level requirements is said to be "unconditionally compliant"; one that satisfies all the MUST level requirements but not all the SHOULD level requirements is said to be "conditionally compliant".

2. Direct Cache Manipulation Extensions

This section specifies Cache-Control request header extensions which allow specification of cache operations, as allowed for in HTTP/1.1[1].

These extensions are intended for the manipulation of shared caches; they MUST NOT be interpreted by non-shared caches.

Upon receiving one of these extension headers, an implementation will perform the operation specified, and then MUST return a response; the type of response returned is dependent on the nature of the operation.

Such responses MUST NOT have an entity body included. If the status code returned would normally have a body, a 'Content-Length: 0'

header MUST be included.

2.1 Prefetch

The prefetch extension allows content providers to insert an object into the cache without incurring the expense of a complete round trip for the object body.

prefetch-extension = "prefetch"

For example, if the resource "http://www.example.com/largeobject" were requested from a DCM-capable shared cache, with a "Cache-Control: prefetch" header, the intermediate would operate exactly as it normally does, except that the response would not contain an entity body.

This avoids the inefficiency of transporting the object both to the cache and then to the device which made the request, which is frequently near the origin server.

Implementations SHOULD support validation, server-driven content negotiation and other HTTP mechanisms in conjunction with this mechanism.

The prefetch extension MUST NOT be forwarded by implementations.

2.2 Eject

This extension enables the eviction of objects from the cache. When an implementation receives this directive, it MUST evict the object identified by the resource, so that the next request for the resource is unconditional (unless the request itself is conditional).

eject-extension = "eject"

When ejecting objects, implementations MUST eject all variants of the resource identified.

Requests containing the eject extension MUST NOT be forwarded by implementations.

3. Security Considerations

This document does not address security-related issues, but the mechanisms it describes should be used in conjuction with appropriate authentication and authorization control. Such control may be implementation-specific, although there are defined mechanisms in the HTTP and elsewhere that may be appropriate.

In particular, unauthorized use of these mechanisms may lead to reduced cache efficiency, denial of service to the intermediate, and may make other security issues in the intermediate more readily exploitable.

References

[1] Fielding, R., Gettys, J., Mogul, J. C., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

Author's Address

Mark Nottingham
Akamai Technologies
Suite 703, 1400 Fashion Island Bvld
San Mateo, CA 94404
US

EMail: mnot@akamai.com
URI: http://www.mnot.net/

Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC editor function is currently provided by the Internet Society.