

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: June 2, 2009

M. Nottingham  
Yahoo! Inc.  
November 29, 2008

**HTTP Cache-Control Extensions for Stale Content**  
**draft-nottingham-http-stale-controls-00**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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 June 2, 2009.

Abstract

This document defines two independent HTTP Cache-Control extensions that allow control over the use of stale responses by caches.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Notational Conventions . . . . .	<a href="#">3</a>
<a href="#">3.</a>	The stale-while-revalidate Cache-Control Extension . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	Example . . . . .	<a href="#">4</a>
<a href="#">4.</a>	The stale-if-error Cache-Control Extension . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	Example . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">6</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">7.</a>	Normative References . . . . .	<a href="#">6</a>
<a href="#">Appendix A.</a>	Acknowledgements . . . . .	<a href="#">6</a>
	Author's Address . . . . .	<a href="#">6</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">8</a>

Nottingham

Expires June 2, 2009

[Page 2]

## **1. Introduction**

HTTP [[RFC2616](#)] requires that caches "respond to a request with the most up-to-date response held... that is appropriate to the request," although "in carefully controlled circumstances" a stale response is allowed to be returned. This document defines two independent Cache-Control extensions that allow for such control, stale-if-error and stale-while-revalidate.

The stale-if-error HTTP Cache-Control extension allows a cache to return a stale response when an error -- e.g., a 500 Internal Server Error, a network segment, or DNS failure -- is encountered, rather than returning a "hard" error. This improves availability.

The stale-while-revalidate HTTP Cache-Control extension allows a cache to immediately return a stale response while it revalidates it in the background, thereby hiding latency (both in the network and on the server) from clients.

## **2. Notational Conventions**

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](#) [[RFC2119](#)].

This specification uses the augmented Backus-Naur Form of [RFC2616](#) [[RFC2616](#)], and includes the delta-seconds rule from that specification.

## **3. The stale-while-revalidate Cache-Control Extension**

When present in an HTTP response, the stale-while-revalidate Cache-Control extension indicates that caches MAY serve the response it appears in after it becomes stale, up to the indicated number of seconds.

stale-while-revalidate = "stale-while-revalidate" "=" delta-seconds

If a cached response is served stale due to the presence of this extension, the cache SHOULD attempt to revalidate it while still serving stale responses (i.e., without blocking).

Note that "stale" implies that the response will have a non-zero Age header and a warning header, as per HTTP's requirements.

If delta-seconds passes without the cached entity being revalidated,

Nottingham

Expires June 2, 2009

[Page 3]

it SHOULD NOT continue to be served stale, absent other information.

### **3.1. Example**

A response containing:

```
Cache-Control: max-age=600, stale-while-revalidate=30
```

indicates that it is fresh for 600 seconds, and it may continue to be served stale for up to an additional 30 seconds while an asynchronous validation is attempted. If validation is inconclusive, or if there is not traffic that triggers it, after 30 seconds the stale-while-revalidate function will cease to operate, and the cached response will be "truly" stale (i.e., the next request will block and be handled normally).

Generally, servers will want to set the combination of max-age and stale-while-revalidate to the longest total potential freshness lifetime that they can tolerate. For example, with both set to 600, the server must be able to tolerate the response being served from cache for up to 20 minutes.

Since asynchronous validation will only happen if a request occurs after the response has become stale, but before the end of the stale-while-revalidate window, the size of that window and the likelihood of a request during it determines how likely it is that all requests will be served without delay. If the window is too small, or traffic too sparse, some requests will fall outside of it, and block until the server can validate the cached response.

## **4. The stale-if-error Cache-Control Extension**

The stale-if-error Cache-Control extension indicates that when an error is encountered, a cached stale response MAY be used to satisfy the request, regardless of other freshness information.

```
stale-if-error = "stale-if-error" "=" delta-seconds
```

When used as a request Cache-Control extension, its scope of application is the request it appears in; when used as a response Cache-Control extension, its scope is any request applicable to the cached response it occurs in.

Its value indicates the upper limit to staleness; when the cached response is more stale than the indicated amount, the cached response SHOULD NOT be used to satisfy the request, absent other information.

Nottingham

Expires June 2, 2009

[Page 4]

In this context, an error is any situation which would result in a 500, 502, 503 or 504 HTTP response status code being returned.

Note that this directive does not affect freshness; stale cached responses that are used SHOULD still be visibly stale when sent (i.e., have a non-zero Age header and a warning header, as per HTTP's requirements.).

#### [4.1.](#) Example

A response containing:

```
HTTP/1.1 200 OK
Cache-Control: max-age=600, stale-if-error=1200
Content-Type: text/plain
```

success

indicates that it is fresh for 600 seconds, and that it may be used if an error is encountered after becoming stale for an additional 1200 seconds.

Thus, if the cache attempts to validate 900 seconds afterwards and encounters:

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/plain
```

failure

the successful response can be returned instead:

```
HTTP/1.1 200 OK
Cache-Control: max-age=600, stale-if-error=1200
Age: 900
Content-Type: text/plain
```

success

After the age is greater than 1800 seconds (i.e., it has been stale for 1200 seconds), the cache must write the error message through.

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/plain
```

failure





## **5. Security Considerations**

The stale-while-revalidate extension provides origin servers with a mechanism for dictating that stale content should be served from caches under certain circumstances, with the expectation that the cached response will be revalidated in the background. It is suggested that such validation be predicated upon an incoming request, to avoid the possibility of an amplification attack (as can be seen in some other pre-fetching and automatic refresh mechanisms). Cache implementers should keep this in mind when deciding the circumstances under which they will generate a request that is not directly initiated by a user or client.

The stale-if-error provides origin servers and clients a mechanism for dictating that stale content should be served from caches under certain circumstances, and does not pose additional security considerations over those of [RFC2616](#), which also allows stale content to be served.

## **6. IANA Considerations**

This document has no actions for IANA.

## **7. Normative References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.

## **[Appendix A](#). Acknowledgements**

Thanks to Ben Drees, John Nienart, Henrik Nordstrom, Evan Torrie, and Chris Westin for their suggestions. The author takes all responsibility for errors and omissions.

Nottingham

Expires June 2, 2009

[Page 6]

Author's Address

Mark Nottingham  
Yahoo! Inc.

Email: [mnot@yahoo-inc.com](mailto:mnot@yahoo-inc.com)  
URI: <http://www.mnot.net/>

## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

Nottingham

Expires June 2, 2009

[Page 8]