Network Working Group P. McManus
Internet-Draft
Mozilla
Intended status: Standards Track October 26, 2016
Expires: April 29, 2017

HTTP Immutable Responses
draft-mcmanus-immutable-00

Abstract

The immutable HTTP response Cache-Control extension allows servers to identify resources that will not be updated during their freshness lifetime. This assures that a client never needs to revalidate a cached fresh resource to be certain it has not been modified.

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 http://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 April 29, 2017.

Copyright Notice

1. Introduction

The HTTP freshness lifetime [RFC7234] caching attribute specifies that a client may safely reuse a response to satisfy future requests over a specific period of time. It does not specify that the resource will be not be modified during that period.

For instance, a front page newspaper photo with a freshness lifetime of one hour would mean that no user should see a photo more than one hour old. However, the photo could be updated at any time resulting in different users seeing different photos depending on the contents of their caches for up to one hour. This is compliant with the caching mechanism defined in [RFC7234].

Users that need to confirm there have been no updates to their current cached resources typically invoke the reload (or refresh) mechanism in the user agent. This in turn generates a conditional request [RFC7232] and either a new representation or, if unmodified, a 304 response [RFC7231] is returned. A user agent that manages HTML and its dependent sub-resources may issue hundreds of conditional requests to refresh all portions of a common HTML page [REQPERPAGE].

Through the use of the versioned URL design pattern some content providers never create more than one variant of a sub-resource. When these resources need an update they are simply published under a new URL, typically embedding a variant identifier in the path, and references to the sub-resource are updated with the new path information.

For example, https://www.example.com/101016/main.css might be updated and republished as https://102026/main.css and the html that references it is changed at the same time. This design pattern allows a very large freshness lifetime to be applied to the sub-resource without guessing when it will be updated in the future.

Unfortunately, the user-agent is not aware of the versioned URL design pattern. User driven refresh events still translate into wasted conditional requests for each sub-resource as each will return 304 responses.

The immutable HTTP response Cache-Control extension allows servers to identify resources that will not be updated during their freshness lifetime. This effectively instructs the client that any conditional request for a previously served variant of that resource may be safely skipped without worrying that it has been updated.

2. The immutable Cache-Control extension

When present in an HTTP response, the immutable Cache-Control extension indicates that the origin server MUST NOT update the representation of that resource during the freshness lifetime of the response.

The immutable extension only applies during the freshness lifetime of the response. Stale responses SHOULD be revalidated as they normally would be in the absence of immutable.

The immutable extension takes no arguments and if any arguments are present they have no meaning. Multiple instances of the immutable extension are equivalent to one instance. The presence of an immutable Cache-Control extension in a request has no effect.

2.1. Example

Cache-Control: max-age=31536000, immutable

3. Security Considerations

The immutable mechanism acts as form of soft pinning and, as with all pinning mechanisms, creates a vector for the amplification of a cache poisoning attack. Two mechanisms are suggested for mitigation of this risk:

- Clients should ignore immutable for resources that are not part of a secure context [SECURECONTEXTS]. Authenticated resources are less vulnerable to cache poisoning.

- User-Agents often provide two different refresh mechanisms: reload and some form of force-reload. The latter is used to rectify interrupted loads and other corruption. These reloads should ignore immutable as well.

4. IANA Considerations

[RFC7234] sections 7.1 and 7.1.2 require registration of the immutable extension in the "Hypertext Transfer Protocol (HTTP) Cache Directive Registry" with IETF Review.

- Cache-Directive: immutable

- Pointer to specification text: [this document]

5. Acknowledgments

Thank you to Ben Maurer for partnership in developing and testing this idea.

6. References

6.1. Normative References

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

[RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <http://www.rfc-editor.org/info/rfc7232>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <http://www.rfc-editor.org/info/rfc7234>.

6.2. Informative References

[SECURECONTEXTS] West, M., "Secure Contexts", n.d., <https://w3c.github.io/webappsec-secure-contexts/>.

[REQPERPAGE] "HTTP Archive", n.d., <http://httparchive.org/interesting.php#reqTotal>.

Author's Address

Patrick McManus
Mozilla

Email: pmcmanus@mozilla.com