Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 4, 2017

M. Nottingham
August 3, 2016

# Site-Wide HTTP Headers
# draft-nottingham-site-wide-headers-00

Abstract

This document specifies an alternative way for Web sites to send HTTP response header fields that apply to large numbers of resources, to improve efficiency.

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 February 4, 2017.

Table of Contents



## 1. Introduction

HTTP response headers are being used for an increasing amount of metadata that applies to an entire site, or large portions of it.

For example, "Strict-Transport-Security" [RFC6797] and "Public-Key-Pins" [RFC7469] both define headers that are explicitly scoped to an entire origin [RFC6454], and number of similar headers are under consideration.

Likewise, some HTTP header fields only sensibly have a single value per origin; for example, "Server".

Furthermore, some headers are used uniformly across an origin. For example, a site might have a "Content-Security-Policy" [W3C.CR-CSP2-20150721] header that doesn't vary across the site, or only varies slightly from resource to resource.

HTTP/2's HPACK [RFC7541] header compression mechanism was designed to reduce bandwidth usage for often-repeated headers, both in responses and requests. However, it limits the amount of compression contents usable for a connection (by default, 4K), which sites are beginning to exceed, thereby reducing the efficiency of HPACK itself.

For example, it is not uncommon for a CSP response header field to exceed 1K (and has been observed to be greater than 3K on popular sites). This forces site administrators to make an awkward choice; put the large header in the HPACK table, thereby crowding out other headers, or omit it, requiring its full content to be sent on every applicable response.

This document defines a way to specify one or more sets of HTTP response header fields in a well-known resource [RFC5785] that, when their use is negotiated, are appended to HTTP responses by the user agent. This allows common response headers to be omitted both from on-the-wire responses and the HPACK compression table, making both more efficient.

This approach is preferable to increasing the HTTP/2 SETTINGS_HEADER_TABLE_SIZE ([RFC7540], Section 6.5.2), because increasing that setting incurs a per-connection overhead on the server, whereas using the technique documented here does not.

### 1.1. Example

If a user agent has a fresh copy of the well-known resource for an origin (see Section 4), because either it performed a GET, or HTTP/2 Server Push was used:

```
HTTP/1.1 200 OK
Content-Type: text/site-headers
Cache-Control: max-age=3600
ETag: "abc123"
Content-Length: 1234

# a
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
```

and the user agent makes the request:

```
GET /images/foo.jpg HTTP/1.1
Host: www.example.com
SM: "abc123"
```

this indicates that the user agent has processed the well-known resource, and therefore that the server can omit the nominated

response header fields on the wire, instead referring to them with the "HS" response header field:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Vary: SM, Accept-Encoding
Cache-Control: max-age=3600
HS: "a"
Transfer-Encoding: chunked
```

Upon receipt of that response, the user agent will consider it equivalent to:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
Vary: SM, Accept-Encoding
Cache-Control: max-age=3600
Connection: close
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
```

If a request omits the "SM" header field, or its field-value does not match the current ETag of the well-known resource, all of the header fields above will be sent by the server in the response.

### 1.2. 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 [RFC2119].

This document uses the following ABNF rules from [RFC5234]: "DQUOTE", "ALPHA". From [RFC7230]: "OWS", "RWS", "CRLF", "header-field". From [RFC7232]: "entity-tag".

## 2. Server Operation

When a server wishes to use site-wide HTTP headers, it places a file in the format specified in Section 4.1 at the well-known URI specified in Section 4.

Then, when a request has a "SM" request header field (as per Section 3.1) that matches the current ETag of the well-known resource, the set of response header fields referred to by the "HS"

response header field (see Section 2.2) for the requested resource are omitted from the corresponding response.

Servers SHOULD include "SM" in the field-value of the "Vary" response header field for all cacheable (as per [RFC7234]) responses of resources that behave in this manner, whether or not headers have been actually appended. This assures correct cache operation, and also advertises support for this specification.

Servers MAY use HTTP/2 Server Push ([RFC7540], Section 8.2) to proactively send the well-known resource to user agents (e.g., if they emit "SM: *", indicating that they do not have a fresh copy of the well-known resource).

## 2.1. Selecting Site-Wide Headers

Because this mechanism effectively hides response header fields from intermediaries that do not implement it, care ought to be take in selecting the headers to use it upon.

For example, the "Cache-Control" and "Vary" headers are poor candidates, because they are often used by intermediaries for HTTP caching [RFC7234].

Likewise, HTTP/1 headers that affect message framing and connection behaviour (e.g., "Content-Length", "Transfer-Encoding", "Connection") MUST NOT be included in the well-known resource.

## 2.2. The "HS" HTTP Response Header Field

The "HS" HTTP response header field indicates the header set in the well-known location file (see Section 4.1) that should be applied to the response it occurs within.

HS = DQUOTE 1*ALPHA DQUOTE

For example:

HS: "foo"

# 3. User Agent Operation

User agents that support this specification SHOULD always emit a "SM" header field in requests, carrying either the "ETag" of the well-known resource currently held for the origin, or "*" to indicate that they support this specification, but do not have a fresh (as per [RFC7234]) copy of it.

User agents might discover that an origin supports this specification when it returns a response containing the "HS" response header field, or they might learn of it when the well-known location's current contents are sent via a HTTP/2 Server Push.

In either case, user agents SHOULD send a "SM" request header field on all requests to such an origin.

Upon receiving a response to such a request containing the "HS" response header field, user agents MUST locate the header-set referred to by its field-value in the stored well-known response, remove any surrounding white space, and append it to the response headers, stripping the "HS" response header field.

If the corresponding header-set cannot be found in the well-known location, the response MUST be considered invalid and MUST NOT be used; the user agent MAY retry the request without the "SM" request header field if its method was safe, or may take alternative recovery strategies.

### 3.1. The "SM" HTTP Request Header Field

The "SM" HTTP request header field indicates that the user agent has a fresh (as per [RFC7234]) copy of the well-known resource (see Section 4) for the request's origin ([RFC6454]).

SM = "*" / entity-tag

Its value is the "entity-tag" [RFC7232] of the freshest valid well-known location response held by the user agent. If none is held, it should be "*" (without quotes).

For example:

SM: "abc123"
SM: *

## 4. The "site-headers" well-known URI

The well-known URI [RFC5785] "site-headers" is a resource that, when fetched, returns a file in the "text/site-headers" format (see Section 4.1).

Its media type SHOULD be generated as "text/site-headers", although user agents SHOULD NOT reject responses with other types (particularly, "application/octet-stream" and "text/plain").

Its representation MUST contain an "ETag" response header [RFC7232].

User agents SHOULD consider it to be valid for its freshness lifetime (as per [RFC7234]). If it does not have an explicit freshness lifetime, they SHOULD consider it to have a heuristic freshness lifetime of 60 seconds.

### 4.1. The "text/site-headers" Media Type

The "text/site-headers" media type is used to indicate that a file contains one or more sets of HTTP header fields, as defined in [RFC7230], Section 3.

```
site-headers = 1*( header-header header-set )
header-header = "#" 1*RWS set-name OWS CRLF
set-name = 1*ALPHA
header-set = OWS *( header-field CRLF ) OWS
```

Each set of HTTP header fields is started by a header-header, which is indicated by an octothorp ("#") followed by the name of the header set. The following lines, up until the next line beginning with an octothorp or the end of the file are considered to be the header-set's contents.

As in HTTP itself, implementations need to be forgiving about line endings; specifically, bare CR MUST be considered to be a line ending.

For example:

```
# foo
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
# bar
Strict-Transport-Security: max-age=15768000 ; includeSubDomains
Server: Apache/2.4.7 (Ubuntu)
Public-Key-Pins: max-age=604800;
  pin-sha256="ZitlqPmA9wodcxkwOW/c7ehlNFk8qJ9FsocodG6GzdjNM=";
  pin-sha256="XRXP987nz4rd1/gS2fJSNVfyrZbqa00T7PeRXUPd15w=";
  report-uri="/lib/key-pin.cgi"
Content-Security-Policy: default-src 'self'; img-src 'self'
  *.staticflickr.com; frame-ancestors 'none';
  report-uri https://mnot.report-uri.io/r/default/csp/enforce
```

This file specifies two sets of HTTP headers, "foo" and "bar". Note that the "Public-Key-Pins" and "Content-Security-Policy" header

fields are line-folded; as in HTTP, this form of header is deprecated in this format, and SHOULD NOT be used (except in documentation, as we see here).

#### 4.1.1. Parsing "text/site-headers"

Given a stream of Unicode characters:

1. Let "header-sets" be an empty mapping.

2. Consume all characters from up to and including the first octothorp ("#").

3. Consume all "WSP" characters.

4. Let "set-name" be all characters up to but not including the next "WSP", "CR" or "LF".

5. Consume all "WSP", "CR" and "LF characters".

6. Let "header-set" be all characters up to but not including the next "CR" or "LF" character followed by an octothorp ("#"), or the end of the file.

7. Trim all "WSP" from the end of "header-set".

8. Let the value of the "set-name" entry in "header-sets" be "header-set" (removing any existing value).

9. If there is more "input", return to step 2.

10. Otherwise, return "header-sets".

This returns a mapping of "set-name" to a HTTP "header-set", as defined in [RFC7230], Section 3. It SHOULD be parsed as defined there.

## 5. IANA Considerations

TBD

## 6. Security Considerations

### 6.1. Injecting Headers

Site-wide headers allow a single resource to inject HTTP response headers for an entire origin. Accordingly, the ability to write to that resource needs to be carefully controlled by the origin server.

### 6.2. Inappropriate Headers

As noted in Section 2.1, there are a variety of HTTP response headers which are inappropriate for use as site-wide headers, and some (e.g., "Content-Length") can cause both interoperability and security issues.

### 6.3. Differing Views of Headers

Because headers sent via this mechanism will not be seen by user agents and intermediaries that do not implement this specification, they will potentially have a different view of the response headers.

## 7. References

### 7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <http://www.rfc-editor.org/info/rfc5785>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <http://www.rfc-editor.org/info/rfc6454>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[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>.

## 7.2. Informative References

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <http://www.rfc-editor.org/info/rfc6797>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <http://www.rfc-editor.org/info/rfc7469>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <http://www.rfc-editor.org/info/rfc7540>.

[RFC7541] Peon, R. and H. Ruellan, "HPACK: Header Compression for HTTP/2", RFC 7541, DOI 10.17487/RFC7541, May 2015, <http://www.rfc-editor.org/info/rfc7541>.

[W3C.CR-CSP2-20150721] West, M., Barth, A., and D. Veditz, "Content Security Policy Level 2", World Wide Web Consortium CR CR-CSP2-20150721, July 2015, <http://www.w3.org/TR/2015/CR-CSP2-20150721>.

## Author's Address

Mark Nottingham

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