Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 6, 2019

M. Nottingham
June 4, 2019

The "safe" HTTP Preference
draft-nottingham-safe-hint-10

Abstract

This specification defines a "safe" preference for HTTP requests that expresses a desire to avoid objectionable content, according to the definition of that term by the origin server.

Support for this preference by clients and servers is optional.

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 December 6, 2019.

Copyright Notice

Table of Contents

1. Introduction

Many Web sites have a "safe" mode, to assist those who don't want to be exposed (or have their children exposed) to content to which they might object.

However, that goal is often difficult to achieve, because of the need to go to every Web site that might be used, navigate to the appropriate page (possibly creating an account along the way) to get a cookie [RFC6265] set in the browser, for each browser on every device used.

A more manageable approach is for the browser to proactively indicate a preference for safe content. A user agent that supports doing so (whether it be an individual browser, or through an Operating System HTTP library) need only be configured once to assure that the preference is advertised to a set of sites, or even all sites.

This specification defines how to declare this desire in requests as a HTTP Preference [RFC7240].

Note that this specification does not define what content might be considered objectionable, and so the concept of "safe" is also not precisely defined. Rather, the term is interpreted by the server and within the scope of each Web site that chooses to act upon this information.

That said, the intent of "safe" is to allow end users (or those acting on their behalf) to express a desire to avoid content that is considered objectionable within the cultural context of that site;

usually (but not always) content that is unsuitable for minors. The "safe" preference is not intended to be used for other purposes.

Furthermore, sending "safe" does not guarantee that the Web site will use it, nor that it will apply a concept of "objectionable" that is

consistent with the requester's views. As such, its effect can be described as "best effort," and not to be relied upon. In other words, sending the preference is no more reliable than going to each Web site and manually selecting a "safe" mode, but it is considerably easier.

It is also important to note that the "safe" preference is not a reliable indicator that the end user is a child; other users might have a desire for unobjectionable content, and some children might browse without the preference being set.

Note also that the cultural context applies to the hosting location of a site, the content provider, and the source of the content. It cannot be guaranteed that a user-agent and origin server will have the same view of the concept of what is objectionable.

Simply put, it is a statement by (or on behalf of) the end user to the effect "If your site has a 'safe' setting, this user is hereby opting into that, according to your definition of the term."

The mechanism described in this document does not have IETF consensus and is not a standard. It is a widely deployed approach that has turned out to be useful, and is presented here so that server and browser implementations can have a common understanding of how it operates.

This mechanism was presented for publication as an IETF Proposed Standard, but was not approved for publication by the IESG despite having IETF consensus at that time. Concerns raised by the IESG included the vagueness of the meaning of "safe", the ability of a proxy to insert the hint outside of a user's control, and the fact that there is no way to know whether the hint was or was not applied to the response returned by the server. While the current text is clear about these issues, they remain as factors that block the IESG's approval of this mechanism as an IETF Proposed Standard.

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 "safe" Preference

When present in a request, the "safe" preference indicates that the user prefers that the origin server to not respond with content which is designated as objectionable, according to the origin server's definition of the concept.

For example, a request that includes the "safe" preference:

```
GET /foo.html HTTP/1.1
Host: www.example.org
User-Agent: ExampleBrowser/1.0
Prefer: safe
```

Typically, user agents that emit the "safe" preference will include it in all requests with the "https" URI scheme, although some might expose finer-grained controls over when it is sent; this ensures that the preference is available to the applicable resources. User agents MUST NOT emit the "safe" preference on requests with the "http" URI scheme (see Section 4). See Appendix B for more information about configuring the set of resources "safe" is sent to.

Safe MAY be implemented in common HTTP libraries (e.g., an operating system might choose to insert the preference in requests based upon system-wide configuration).

Origin servers that utilize the "safe" preference ought to document that they do so, along with the criteria that they use to denote objectionable content. If a server has more fine-grained degrees of

"safety", it SHOULD select a reasonable default to use, and document that; it MAY use additional mechanisms (e.g., cookies [RFC6265]) to fine-tune.

A response corresponding to the request above might have headers that look like this:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/html
Preference-Applied: safe
Server: ExampleServer/2.0
Vary: Prefer
```

Here, the Preference-Applied response header ([RFC7240]) indicates that the site has applied the preference. Servers are not required to send Preference-Applied (even when they have applied the preference), but are encouraged to where possible.

Note that the Vary response header needs to be sent if the response is cacheable and might change depending on the value of the "Prefer" header. This is not only true for those responses that are "safe", but also the default "unsafe" response.

See [RFC7234] Section 4.1 for more information the interaction between Vary and Web caching.

See Appendix C for additional advice specific to Web servers wishing to use "safe".

3. Implementation Status

Note to RFC Editor: Please remove this section before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available

implementations or their features. Readers are advised to note that other implementations may exist.

- Microsoft Internet Explorer - see https://support.microsoft.com/en-hk/help/2980016/
- Microsoft Bing - see https://developer.microsoft.com/en-us/microsoft-edge/testdrive/demos/familysearch/
- Mozilla Firefox - see https://support.mozilla.org/en-US/kb/block-and-unblock-websites-parental-controls-firef
- Cisco - see http://blogs.cisco.com/security/filtering-explicit-content

4. Security Considerations

The "safe" preference is not a secure mechanism; it can be inserted or removed by intermediaries with access to the request stream (e.g. for "http" URLs). Therefore, it is prohibited from being included in requests with the "http" scheme.

Its presence reveals limited information about the user, which may be of small assistance in "fingerprinting" the user by sites. Therefore, user agents SHOULD NOT include it in requests when the

user has expressed a desire to avoid such attacks (e.g., some forms of "private mode" browsing).

By its nature, including "safe" in requests does not assure that all content will actually be safe; it is only when servers elect to honor it that content might be "safe".

Even then, a malicious server might adapt content so that it is even less "safe" (by some definition of the word). As such, this mechanism on its own is not enough to assure that only "safe" content is seen; those who wish to ensure that will need to combine its use with other techniques (e.g., content filtering).

Furthermore, the server and user may have differing ideas regarding the semantics of "safe." As such, the "safety" of the user's

experience when browsing from site to site as well as over time might (and probably will) change.

5. IANA Considerations

This specification registers the following entry in the "HTTP Preferences" registry [RFC7240]:

- Preference: safe
- Value: (no value)
- Description: Indicates that "safe" / "unobjectionable" content is preferred.
- Reference: (this document)
- Notes:

6. References

6.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[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, <https://www.rfc-editor.org/info/rfc7234>.

[RFC7240] Snell, J., "Prefer Header for HTTP", RFC 7240, DOI 10.17487/RFC7240, June 2014, <https://www.rfc-editor.org/info/rfc7240>.

[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>.

6.2. Informative References

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <https://www.rfc-editor.org/info/rfc6265>.

Appendix A. Acknowledgements

Thanks to Alissa Cooper, Ilya Grigorik, Emma Llanso, Jeff Hughes, Lorrie Cranor, Doug Turner and Dave Crocker for their comments.

Appendix B. Sending "safe" from Web Browsers

As discussed in Section 2, there are many possible ways for the "safe" preference to be generated. One possibility is for a Web browser to allow its users to configure the preference to be sent.

When doing so, it is important not to misrepresent the preference as binding to Web sites. For example, an appropriate setting might be a checkbox with wording such as:

> [] Request "safe" content from Web sites

... along with further information available upon request.

Browsers might also allow the "safe" preference to be "locked" - that is, prevent modification without administrative access, or a passcode.

Note that this specification does not require browsers to send "safe" on all requests, although that is one possible implementation; e.g., alternate implementation strategies include blacklists and whitelists.

Appendix C. Supporting "safe" on Web Sites

Web sites that allow configuration of a "safe" mode (for example, using a cookie) can add support for the "safe" preference incrementally; since the preference will not be supported by all clients immediately, it is necessary to have another way to configure it.

When honoring the safe preference, it is important that it not be possible to disable it through the Web site's interface, since "safe" may be configured and locked down by the browser or computer's administrator (e.g., a parent). If the site has such a means of configuration (e.g., stored user preferences) and the safe preference is received in a request, the "safer" interpretation ought to be used.

The appropriate level of "safety" is a site-specific decision. When selecting it, sites ought to bear in mind that disabling the preference might be considerably more onerous than through other

means, especially if the preference is generated based upon Operating System configuration.

Sites might offer different levels of "safeness" through Web configuration, they will need to either inform their users of what level the "safe" hint corresponds to, or provide them with some means of adjusting it.

If the user expresses a wish to disable "safe" mode, the site can remind them that the safe preference is being sent, and ask them to consult their administrator (since "safe" might be set by a locked-down Operating System configuration).

As explained in Section 2, responses that change based upon the presence of the "safe" preference need to either carry the "Vary: Prefer" response header field, or be uncacheable by shared caches (e.g., with a "Cache-Control: private" response header field). This is to avoid an unsafe cached response being served to a client that prefers safe content (or vice versa).

Author's Address

Mark Nottingham

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