Stream: Independent Submission
RFC: 8674
Category: Informational
Published: December 2019
ISSN: 2070-1721
Author: M. Nottingham

RFC 8674
The "safe" HTTP Preference

Abstract

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

This specification does not define a precise semantic for "safe". Rather, the term is interpreted by the server and within the scope of each web site that chooses to act upon this information.

Support for this preference by clients and servers is optional.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8674.

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 and 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 ensure 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 an HTTP Preference [RFC7240].

Note that this specification does not define what content might be considered objectionable, so the concept of "safe" is 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 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), the objectionable content is content unsuitable for minors. The safe preference is not intended to be used for other purposes.

Furthermore, sending the preference does not guarantee that the web site will use it or 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 indicating that "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 because of concerns that included the vagueness of the meaning of "safe", the ability of a proxy to insert the hint outside of a user's control, the fact that there was no way to know whether the hint was or was not applied to the response returned by the server, and the possibility that the use of this preference may incentivize increased censorship and/or targeting of minors.

The specification was updated to address those concerns, but the IESG did not approve progressing this document as an IETF Proposed Standard. As a result, it has been published in the Independent Stream.

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 not respond with content that is designated as objectionable, according to the origin server's definition of the concept.

For example, this is 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 3). See Appendix A for more information about configuring the set of resources the safe preference is sent to.

The safe preference **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 Section 4.1 of [RFC7234] for more information about the interaction between the Vary header field and web caching.

See Appendix B for additional advice specific to web servers wishing to use the safe preference.

3. 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 information about the user, which may be of assistance in fingerprinting the user by sites and other entities in the network. This information provides insight into the preferences of the user and might be used to make assumptions about user; thus, it could be used to identify categories of users for purposes such as targeting (including advertising and identification of minors). 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 the safe preference in requests does not ensure that all content will actually be safe; content is safe only when servers elect to honor it.

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 ensure 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.

4. IANA Considerations

Per this specification, IANA has registered the following entry in the "HTTP Preferences" registry [RFC7240]:

- Preference: safe
- Description: Indicates that safe (i.e., unobjectionable) content is preferred.
- Reference: RFC 8674

5. References

5.1. 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>.

5.2. 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. Sending the "safe" Preference 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 to prevent modification without administrative access or a passcode.

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

Appendix B. Supporting the "safe" Preference 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 the safe preference 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 using other means, especially if the preference is generated based upon the operating system configuration.

Sites might offer different levels of safety 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 users express 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 the safe preference 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).

Acknowledgements

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

Author's Address

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