Network Working Group
Request for Comments: 2310
Category: Experimental

K. Holtman
TUE

April 1998

The Safe Response Header Field

Status of this Memo

Copyright Notice

Abstract

This document defines a HTTP response header field called Safe, which can be used to indicate that repeating a HTTP request is safe. Such an indication will allow user agents to handle retries of some safe requests, in particular safe POST requests, in a more user-friendly way.

1 Introduction

This document defines a HTTP response header field called Safe, which can be used to indicate that repeating a HTTP request is safe. Such an indication will allow user agents to handle retries of some safe requests, in particular safe POST requests, in a more user-friendly way.

2 Terminology and Notation

This document uses the HTTP terminology and BNF notation defined in [1]. It uses the key words in RFC 2119 for defining the significance of each particular requirement.

3 Rationale

According to Section 9.1.1 (Safe Methods) of the HTTP/1.1 specification [1], POST requests are assumed to be ‘unsafe’ by default. ‘Unsafe’ means ‘causes side effects for which the user will be held accountable’.

It is sometimes necessary for a user agent to repeat a POST request. Examples of such cases are

- when retrying a POST request which gave an indeterminate error result in the previous attempt
- when the user presses the RELOAD button while a POST result is displayed
- when the history function is used to redisplay a POST result which is no longer in the history buffer.

If the POST request is unsafe, HTTP requires explicit user confirmation is before the request is repeated. The confirmation dialog often takes the form of a 'repost form data?' dialog box. This dialog is confusing to many users, and slows down navigation in any case.

If the repeated POST request is safe, the user-unfriendly confirmation dialog can be omitted. However plain HTTP/1.1 [1] has no mechanism by which agents can tell if POST requests are safe, and they must be assumed unsafe by default. This document adds a mechanism to HTTP, the Safe header field, for telling if a POST request is safe.

Using the Safe header field, web applications which require the use of a safe POST request, rather than a GET request, for the submission of web forms, can be made more user-friendly. The use of a POST request may be required for a number of reasons, including

- the contents of the form are potentially very large
- the form is used to upload a file (see [2])
- the application needs some internationalization features (see [3]) which are only available if the form contents are transmitted in a request body the information in the form cannot be encoded in a GET request URL because of security considerations.

4 The Safe response header field

The Safe response header field is defined as an addition to the HTTP/1.x protocol suite.

The Safe response header field is used by origin servers to indicate whether repeating the received HTTP request is safe in the sense of Section 9.1.1 (Safe Methods) of the HTTP/1.1 specification [1]. For the purpose of this specification, a HTTP request is considered to be a repetition of a previous request if both requests

- are issued by the same user agent, and
- apply to the same resource, and
- have the same request method, and
- both have no request body, or both have request bodies which are byte-wise identical after decoding of any content and transfer codings.

The Safe header field has the following syntax.

```
Safe        = "Safe" ":" safe-nature
safe-nature = "yes" | "no"
```

An example of the header field is:

```
Safe: yes
```

If a Safe header field is absent in the response, the corresponding request MUST be considered unsafe, unless it is a GET or HEAD request. As GET and HEAD requests are safe by definition, user agents SHOULD ignore a ‘Safe: no’ header field in GET and HEAD responses.

If, according to a received Safe header field, the repeating of a request is safe, the request MAY be repeated automatically without asking for user confirmation.

5 Security Considerations

For a discussion of the security considerations connected to HTTP form submission, see [1]. The Safe header field introduces no new security risks.

The use of GET requests for form submission has some security risks which are absent for submission with other HTTP methods. By taking away a counter-incentive to the use of GET requests for form submission, the Safe header field may improve overall security.

6 References

[1] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

[2] Nebel, E., and L. Masinter, "Form-based File Upload in HTML", RFC 1867, November 1995.

[3] Yergeau, F., Nicol, G., Adams, G., and M. Duerst, "Internationalization of the Hypertext Markup Language", RFC 2070, January 1997.

7 Author’s Address

Koen Holtman
Technische Universiteit Eindhoven
Postbus 513
Kamer HG 6.57
5600 MB Eindhoven (The Netherlands)

EMail: koen@win.tue.nl

8. Full Copyright Statement