Internet Engineering Task Force (IETF)
Request for Comments: 8297
Category: Experimental
ISSN: 2070-1721

K. Oku
Fastly

December 2017

An HTTP Status Code for Indicating Hints

Abstract

This memo introduces an informational HTTP status code that can be used to convey hints that help a client make preparations for processing the final response.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate 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/rfc8297.

Table of Contents

1. Introduction

It is common for HTTP responses to contain links to external resources that need to be fetched prior to their use, for example, rendering HTML by a web browser. Having such links available to the client as early as possible helps to minimize perceived latency.

The "preload" [Preload] link relation can be used to convey such links in the Link header field of an HTTP response. However, it is not always possible for an origin server to generate the header block of a final response immediately after receiving a request. For example, the origin server might delegate a request to an upstream HTTP server running at a distant location, or the status code might depend on the result of a database query.

The dilemma here is that even though it is preferable for an origin server to send some header fields as soon as it receives a request, it cannot do so until the status code and the full header fields of the final HTTP response are determined.

HTTP/2 [RFC7540] server push can accelerate the delivery of resources, but only resources for which the server is authoritative. The other limitation of server push is that the response will be transmitted regardless of whether the client has the response cached. At the cost of spending one extra round trip compared to server push in the worst case, delivering Link header fields in a timely fashion is more flexible and might consume less bandwidth.

This memo defines a status code for sending an informational response ([RFC7231], Section 6.2) that contains header fields that are likely to be included in the final response. A server can send the informational response containing some of the header fields to help the client start making preparations for processing the final response, and then run time-consuming operations to generate the

final response. The informational response can also be used by an origin server to trigger HTTP/2 server push at a caching intermediary.

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. HTTP Status Code 103: Early Hints

The 103 (Early Hints) informational status code indicates to the client that the server is likely to send a final response with the header fields included in the informational response.

Typically, a server will include the header fields sent in a 103 (Early Hints) response in the final response as well. However, there might be cases when this is not desirable, such as when the server learns that the header fields in the 103 (Early Hints) response are not correct before the final response is sent.

A client can speculatively evaluate the header fields included in a 103 (Early Hints) response while waiting for the final response. For example, a client might recognize a Link header field value containing the relation type "preload" and start fetching the target resource. However, these header fields only provide hints to the client; they do not replace the header fields on the final response.

Aside from performance optimizations, such evaluation of the 103 (Early Hints) response's header fields MUST NOT affect how the final response is processed. A client MUST NOT interpret the 103 (Early Hints) response header fields as if they applied to the informational response itself (e.g., as metadata about the 103 (Early Hints) response).

A server MAY use a 103 (Early Hints) response to indicate only some of the header fields that are expected to be found in the final response. A client SHOULD NOT interpret the nonexistence of a header field in a 103 (Early Hints) response as a speculation that the header field is unlikely to be part of the final response.

The following example illustrates a typical message exchange that involves a 103 (Early Hints) response.

Client request:

```
GET / HTTP/1.1
Host: example.com
```

Server response:

```
HTTP/1.1 103 Early Hints
Link: </style.css>; rel=preload; as=style
Link: </script.js>; rel=preload; as=script

HTTP/1.1 200 OK
Date: Fri, 26 May 2017 10:02:11 GMT
Content-Length: 1234
Content-Type: text/html; charset=utf-8
Link: </style.css>; rel=preload; as=style
Link: </script.js>; rel=preload; as=script

<!doctype html>
[... rest of the response body is omitted from the example ...]
```

As is the case with any informational response, a server might emit more than one 103 (Early Hints) response prior to sending a final response. This can happen, for example, when a caching intermediary generates a 103 (Early Hints) response based on the header fields of a stale-cached response, and then forwards a 103 (Early Hints) response and a final response that were sent from the origin server in response to a revalidation request.

A server MAY emit multiple 103 (Early Hints) responses with additional header fields as new information becomes available while the request is being processed. It does not need to repeat the fields that were already emitted, though it doesn’t have to exclude them either. The client can consider any combination of header fields received in multiple 103 (Early Hints) responses when anticipating the list of header fields expected in the final response.

The following example illustrates a series of responses that a server might emit. In the example, the server uses two 103 (Early Hints) responses to notify the client that it is likely to send three Link header fields in the final response. Two of the three expected header fields are found in the final response. The other header field is replaced by another Link header field that contains a different value.

```
HTTP/1.1 103 Early Hints
Link: </main.css>; rel=preload; as=style

HTTP/1.1 103 Early Hints
Link: </style.css>; rel=preload; as=style
Link: </script.js>; rel=preload; as=script

HTTP/1.1 200 OK
Date: Fri, 26 May 2017 10:02:11 GMT
Content-Length: 1234
Content-Type: text/html; charset=utf-8
Link: </main.css>; rel=preload; as=style
Link: </newstyle.css>; rel=preload; as=style
Link: </script.js>; rel=preload; as=script

<!doctype html>
[... rest of the response body is omitted from the example ...]
```

3. Security Considerations

Some clients might have issues handling a 103 (Early Hints) response, because informational responses are rarely used in reply to requests not including an Expect header field ([RFC7231], Section 5.1.1).

In particular, an HTTP/1.1 client that mishandles an informational response as a final response is likely to consider all responses to the succeeding requests sent over the same connection to be part of the final response. Such behavior might constitute a cross-origin information disclosure vulnerability in case the client multiplexes requests to different origins onto a single persistent connection.

Therefore, a server might refrain from sending 103 (Early Hints) responses over HTTP/1.1 unless the client is known to handle informational responses correctly.

HTTP/2 clients are less likely to suffer from incorrect framing since handling of the response header fields does not affect how the end of the response body is determined.

4. IANA Considerations

The following entry has been registered in the "HTTP Status Codes" registry:

- Code: 103
- Description: Early Hints
- Specification: RFC 8297 (this document)

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>.

[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, <https://www.rfc-editor.org/info/rfc7231>.

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

[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

[Preload] Grigorik, I., Ed. and Y. Weiss, Ed., "Preload", W3C Candidate Recommendation, October 2017, <https://www.w3.org/TR/preload/>.

Acknowledgements

Thanks to Tatsuhiro Tsujikawa for coming up with the idea of sending the Link header fields using an informational response.

Mark Nottingham and Willy Tarreau provided substantial help in clarifying the semantics of the status code.

Early stages of the author's work on this document was supported by DeNA Co., Ltd. during his employment there.

Author's Address

Kazuho Oku
Fastly

Email: kazuhooku@gmail.com