

HTTP Working Group
Internet-Draft
Intended status: Experimental
Expires: May 1, 2018

K. Oku
Fastly
October 28, 2017

**An HTTP Status Code for Indicating Hints
draft-ietf-httpbis-early-hints-05**

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.

Note to Readers

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/> .

Working Group information can be found at <https://httpwg.github.io/> ; source code and issues list for this draft can be found at <https://github.com/httpwg/http-extensions/labels/early-hints> .

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 May 1, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1.](#) Introduction [2](#)
- [1.1.](#) Notational Conventions [3](#)
- [2.](#) 103 Early Hints [3](#)
- [3.](#) Security Considerations [5](#)
- [4.](#) IANA Considerations [6](#)
- [5.](#) References [6](#)
- [5.1.](#) Normative References [6](#)
- [5.2.](#) Informative References [6](#)
- [Appendix A.](#) Changes [6](#)
- [A.1.](#) Since [draft-ietf-httpbis-early-hints-04](#) [6](#)
- [A.2.](#) Since [draft-ietf-httpbis-early-hints-03](#) [7](#)
- [A.3.](#) Since [draft-ietf-httpbis-early-hints-02](#) [7](#)
- [A.4.](#) Since [draft-ietf-httpbis-early-hints-01](#) [7](#)
- [A.5.](#) Since [draft-ietf-httpbis-early-hints-00](#) [7](#)
- [Appendix B.](#) Acknowledgements [7](#)
- Author's Address [7](#)

[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.

Oku

Expires May 1, 2018

[Page 2]

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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

2. 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 they 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).

Oku

Expires May 1, 2018

[Page 3]

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, 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.

Oku

Expires May 1, 2018

[Page 4]

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 103 (Early Hints), since 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 Early Hints 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 HTTP Status Codes Registry will be updated with the following entry:

- o Code: 103
- o Description: Early Hints
- o Specification: [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>>.
- [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, <<https://www.rfc-editor.org/info/rfc7230>>.
- [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>>.

5.2. Informative References

- [Preload] Grigorik, I., "Preload", n.d., <<https://w3c.github.io/preload/>>.

Appendix A. Changes

A.1. Since [draft-ietf-httpbis-early-hints-04](#)

- o Clarified that the server is allowed to add headers not found in a 103 response to the final response.

Oku

Expires May 1, 2018

[Page 6]

- o Clarify client's behavior when it receives more than one 103 response.

A.2. Since [draft-ietf-httpbis-early-hints-03](#)

- o Removed statements that were either redundant or contradictory to [RFC7230](#)-7234.
- o Clarified what the server's expected behavior is.
- o Explain that a server might want to send more than one 103 response.
- o Editorial Changes.

A.3. Since [draft-ietf-httpbis-early-hints-02](#)

- o Editorial changes.
- o Added an example.

A.4. Since [draft-ietf-httpbis-early-hints-01](#)

- o Editorial changes.

A.5. Since [draft-ietf-httpbis-early-hints-00](#)

- o Forbid processing the headers of a 103 response as part of the informational response.

[Appendix B](#). Acknowledgements

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

Author's Address

Kazuho Oku
Fastly

Email: kazuhooku@gmail.com

Oku

Expires May 1, 2018

[Page 7]