An HTTP Status Code for Indicating Hints
draft-ietf-httpbis-early-hints-00

Abstract

   This memo introduces an informational status code for HTTP that can
   be used for indicating hints to help a client start making
   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 August 12, 2017.

Copyright Notice

Table of Contents

1. Introduction

   Most if not all of the web pages processed by a web browser contain
   links to external resources that need to be fetched prior to
   rendering the documents. Therefore, it is beneficial to send such
   links as early as possible in order to minimize the time spent until
   the browser becomes possible to render the document. Link header of
   type "preload" ([Preload]) can be used to indicate such links within
   the response headers of an HTTP response.

   However, it is not always possible for an origin server to send a
   response immediately after receiving a request. In fact, it is often
   the contrary. There are many deployments in which an origin server
   needs to query a database before generating a response. It is also
   not unusual for an origin server to delegate a request to an upstream
   HTTP server running at a distant location.

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

   HTTP/2 ([RFC7540]) push can be used as a solution to the issue, but
   has its own limitations. The resources that can be pushed using
   HTTP/2 are limited to those belonging to the same origin. Also, it

is impossible to send only the links of the resources using HTTP/2
push. Sending HTTP responses for every resource is an inefficient
way of using bandwidth, especially when a caching server exists as an
intermediary.

This memo defines a status code for sending an informational response
([RFC7231], section 6.2) that contains headers that are likely to be
included in the final response. A server can send the informational
response containing some of the headers 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 push at an 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

This informational status code indicates the client that the server
is likely to send a final response with the headers included in the
informational response.

A server MUST NOT include Content-Length, Transfer-Encoding, or any
hop-by-hop headers ([RFC7230], section 6.1) in the informational
response using the status code.

A client MAY speculatively evaluate the headers included in the
informational response while waiting for the final response. For
example, a client may recognize the link header of type preload and
start fetching the resource. However, the evaluation MUST NOT affect
how the final response is processed; the client must behave as if it
had not seen the informational response.

An intermediary MAY drop the informational response. It MAY send
HTTP/2 ([RFC7540]) push responses using the information found in the
informational response.

3. Security Considerations

Clients may have issues handling Early Hints, since informational
response is rarely used for requests not including an Expect header
([RFC7231], section 5.1.1).

An HTTP/1.1 client that mishandles the informational response as a
final response is likely to consider all the responses to the
succeeding requests sent over the same connection to be part of the
final response. Such behavior may 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 when 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 headers does not affect how the end of the
response body is determined.

4. IANA Considerations

If Early Hints is standardized, the HTTP Status Codes Registry should
be updated with the following entries:

o Code: 103

o Description: Early Hints

o Specification: this document

5. Acknowledgements

Thanks to Tatsuhiro Tsujikawa for coming up with the idea of sending
the link headers using an informational response.

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,
            <http://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,
            <http://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,
              <http://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,
              <http://www.rfc-editor.org/info/rfc7540>.

6.2. Informative References

   [Preload] Grigorik, I., "Preload", September 2016,
              <https://w3c.github.io/preload/>.

Author's Address

   Kazuho Oku
   DeNA Co., Ltd.

   Email: kazuhooku@gmail.com