HTTP
Internet-Draft
Updates: ietf-httpbis-client-hints (if approved)
Intended status: Experimental
Expires: 25 August 2025

V. Tan
Google LLC

21 February 2025

Critical-CH for Client Hint Reliability
draft-victortan-httpbis-chr-critical-ch-00

Abstract

This document defines the Critical-CH HTTP response header to allow HTTP servers to reliably specify their Client Hint preferences.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/Tanych/http-client-hint-reliability.

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 https://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 25 August 2025.

Copyright Notice

Table of Contents

1. Introduction

[RFC8942] defines a response header, Accept-CH, for servers to advertise a set of request headers used for proactive content negotiation. This allows user agents to send request headers only when used, improving their performance overhead as well as reducing passive fingerprinting surface.

However, on the first HTTP request to a server, the user agent will not have received the Accept-CH header and may not take the server preferences into account. More generally, the server's configuration may have changed since the most recent HTTP request to the server. This document addresses this issue by introducing an HTTP response header, Critical-CH, for the server to instruct the user agent to retry the request if client hints were not initially included but would have been sent based on the received Accept-CH header.

2. Conventions and Definitions

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.

This document uses the Augmented Backus-Naur Form (ABNF) notation of [RFC5234].

This document uses the variable-length integer encoding and frame diagram format from [RFC9000].

3. The Critical-CH Response Header Field

When a user agent requests a resource based on a missing or outdated Accept-CH value, it may not send a desired request header field. Neither user agent nor server has enough information to reliably and efficiently recover from this situation. The server can observe that the header is missing, but the user agent may not have supported the header, or may have chosen not to send it. Triggering a new request in these cases would risk an infinite loop or an unnecessary round-trip.

Conversely, the user agent can observe that a request header appears in the Accept-CH (Section 3.1 of [RFC8942]) and Vary (Section 7.1.4 of [RFC7231]) response header fields. However, retrying based on this information would waste resources if the resource only used the Client Hint as an optional optimization.

This document introduces critical Client Hints. These are the Client Hints which meaningfully change the resulting resource. For example, a server may use the Device-Memory Client Hint [DEVICE-MEMORY] to select simple and complex variants of a resource to different user agents. Such a resource should be fetched consistently across page loads to avoid jarring user-visible switches.

The server specifies critical Client Hints with the Critical-CH response header field. It is a Structured Header [RFC8941] whose value MUST be an sf-list (Section 3.1 of [RFC8941]) whose members are tokens (Section 3.3.4 of [RFC8941]). Its ABNF is:

```
Critical-CH = sf-list
```

For example:

```
Critical-CH: Sec-CH-Example, Sec-CH-Example-2
```

Each token listed in the Critical-CH header SHOULD additionally be present in the Accept-CH and Vary response headers.

When a user agent receives an HTTP response containing a Critical-CH header, it first processes the Accept-CH header as described in Section 3.1 of [RFC8942]. It then performs the following steps:

1. If the request did not use a safe method (Section 4.2.1 of [RFC7231]), ignore the Critical-CH header and continue processing the response as usual.

2. If the response was already the result of a retry, ignore the Critical-CH header and continue processing the response as usual.

3. Determine the Client Hints that would have been sent given the updated Accept-CH value, incorporating the user agent's local policy and user preferences. See also Section 2.1 of [RFC8942].

4. Compare this result to the Client Hints which were sent. If any Client Hint listed in the Critical-CH header was not previously sent and would now have been sent, retry the request with the new preferences. Otherwise, continue processing the response as usual.

Note this procedure does not cause the user agent to send Client Hints it would not otherwise send.

3.1. Example

For example, if the user agent loads https://example.com with no knowledge of the server's Accept-CH preferences, it may send the following response:

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

HTTP/1.1 200 OK
Content-Type: text/html
Accept-CH: Sec-CH-Example, Sec-CH-Example-2
Vary: Sec-CH-Example
Critical-CH: Sec-CH-Example
```

In this example, the server, across the whole origin, uses both Sec-CH-Example and Sec-CH-Example-2 Client Hints. However, this resource only uses Sec-CH-Example, which it considers critical.

The user agent now processes the Accept-CH header and determines it would have sent both headers. Sec-CH-Example is listed in Critical-CH, so the user agent retries the request, and receives a more specific response.

```
GET / HTTP/1.1
Host: example.com
Sec-CH-Example: 1
Sec-CH-Example-2: 2

HTTP/1.1 200 OK
Content-Type: text/html
Accept-CH: Sec-CH-Example, Sec-CH-Example-2
Vary: Sec-CH-Example
```

```
Critical-CH: Sec-CH-Example
```

4. Security Considerations

Request header fields may expose sensitive information about the user's environment. Section 4.1 of [RFC8942] discusses some of these considerations. The document augments the capabilities of Client Hints, but does not change these considerations. The procedure described in Section 3 does not result in the user agent sending request headers it otherwise would not have.

5. IANA Considerations

Features relying on this document are expected to register added request header fields in the Permanent Message Header Fields registry ([RFC3864]).

This document defines the "Critical-CH" HTTP response header field, and registers it in the same registry.

5.1. Critical-CH Header Field

Header field name: Critical-CH

Applicable protocol: HTTP

Status: Standard

Author/Change controller: IETF

Specification document: this specification (Section 3)

Related information: for Client Hints

6. References

6.1. Normative References

[I-D.davidben-http-client-hint-reliability] Benjamin, D., "Client Hint Reliability", Work in Progress, Internet-Draft, draft-davidben-http-client-hint-reliability-03, 1 June 2021, <https://datatracker.ietf.org/doc/html/draft-davidben-http-client-hint-reliability-03>.

[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>.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004, <https://www.rfc-editor.org/info/rfc3864>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[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>.

[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>.

[RFC8941] Nottingham, M. and P. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/info/rfc8941>.

[RFC8942] Grigorik, I. and Y. Weiss, "HTTP Client Hints", RFC 8942, DOI 10.17487/RFC8942, February 2021, <https://www.rfc-editor.org/info/rfc8942>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

6.2. Informative References

[DEVICE-MEMORY] Panicker, S., "Device Memory", n.d., <https://w3c.github.io/device-memory/>.

Acknowledgments

This document has benefited from the work of David Benjamin in [I-D.davidben-http-client-hint-reliability], contributions and suggestions from Ilya Grigorik, Nick Harper, Matt Menke, Aaron Tagliaboschi, Victor Vasiliev, Yoav Weiss, and others.

Author’s Address

Victor Tan
Google LLC
Email: victortan@google.com