

HTTP Client Hints
draft-grigorik-http-client-hints-00

Abstract

An increasing diversity of connected device form factors and software capabilities has created a need to deliver varying, or optimized content for each device.

Client Hints can be used as input to proactive content negotiation; just as the Accept header allowed clients to indicate what formats they prefer, Client Hints allow clients to indicate a list of device and agent specific preferences.

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 September 16, 2013.

Copyright Notice

Copyright (c) 2013 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](#) [3](#)
- [1.1. Notational Conventions](#) [3](#)
- [2. The "CH" Request Header Field](#) [3](#)
- [2.1. Hint Syntax](#) [4](#)
- [2.2. Pre-defined Hints](#) [4](#)
- [2.2.1. dh](#) [4](#)
- [2.2.2. dw](#) [5](#)
- [2.2.3. dpr](#) [5](#)
- [2.3. Examples](#) [5](#)
- [2.4. Server opt-in with Hop and Origin Hints](#) [5](#)
- [2.5. Interaction with Caches](#) [6](#)
- [2.6. Relationship to the User-Agent Request Header](#) [6](#)
- [3. IANA Considerations](#) [6](#)
- [3.1. The CH Request Header Field](#) [6](#)
- [3.2. The HTTP Hints](#) [7](#)
- [3.3. The HTTP Client Hints Registry](#) [7](#)
- [4. Security Considerations](#) [8](#)
- [5. Normative References](#) [8](#)
- [Author's Address](#) [8](#)

1. Introduction

There are thousands of different devices accessing the web, each with different device capabilities and preference information. These device capabilities include hardware and software characteristics, as well as dynamic user and client preferences.

One way to infer some of these capabilities is through User-Agent (UA) detection against an established database of client signatures. However, this technique requires acquiring such a database, integrating it into the serving path, and keeping it up to date. However, even once this infrastructure is deployed, UA sniffing has the following limitations:

- o UA detection requires an external device database
- o UA detection cannot reliably identify all static variables
- o UA detection cannot infer any dynamic client preferences
- o UA detection is not cache friendly

A popular alternative strategy is to use HTTP cookies to communicate some information about the client. However, this approach is also not cache friendly, bound by same origin policy, and imposes additional client-side latency by requiring JavaScript execution to create and manage HTTP cookies.

This document defines a new request Client Hint header field, "CH", that allows the client to perform proactive content negotiation [[I-D.ietf-httpbis-p2-semantic](#)s] by indicating a list of device and agent specific preferences, through a mechanism similar to the Accept header which is used to indicate preferred response formats.

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](#)].

This document uses the Augmented Backus-Naur Form (ABNF) notation of [[RFC5234](#)] with the list rule extension defined in [[I-D.ietf-httpbis-p1-messaging](#)], [Appendix B](#). It includes by reference the OWS, field-name and quoted-string rules from that document, and the parameter rule from [[I-D.ietf-httpbis-p2-semantic](#)s].

2. The "CH" Request Header Field

The "CH" request header field describes an example list of client

Grigorik

Expires September 16, 2013

[Page 3]

preferences that the server can use to adapt and optimize the resource to satisfy a given request. The CH field-value is a comma-delimited list of header fields, and the field-name values are case insensitive.

```
CH = #client-hint
client-hint = parameter
```

[2.1.](#) Hint Syntax

Hints are allowed to have a numeric value. However, where possible, they can be defined as flags (i.e., as a hint name only), so that the hints don't consume too much space in client requests.

Hints can be defined as one of two types:

- o Boolean - indicated by the presence of the hint name. If the hint name is absent in the last message containing the client hint header field, it is considered false.
- o Numeric - value indicated by the digits after "=", up to the first non-digit character. If the hint does not have an argument, its value is assumed to be 0.

Note that HTTP/1.1 allows headers with comma-separated values to be conveyed using multiple instances of the same header; as a result, the hints are collected from all instances of the CH header on the message in question before being considered complete.

[2.2.](#) Pre-defined Hints

The client controls which header fields are communicated within the CH header, based on its default settings, or based on user configuration and preferences. The user may be given the choice to enable, disable, or override specific hints. For example, to allow the request for low-resolution images or other content type's while roaming on a foreign network, even while on a high-bandwidth link.

The client and server, or an intermediate proxy, may use an additional mechanism to negotiate which fields should be reported to allow for efficient content adaptation.

This document defines the following hint names:

[2.2.1.](#) dh

- o Description: device-width in secondary orientation, in density independent pixels.

- o Value Type: number

[2.2.2.](#) dw

- o Description: device-width in primary orientation, in density independent pixels.
- o Value Type: number

[2.2.3.](#) dpr

- o Description: Device Pixel Ratio (dpr), is the ratio between physical pixels and density independent pixels on the device.
- o Value Type: number

Other client hints may be communicated by the client. The decision as to which specific hints will be sent is made by the client.

[2.3.](#) Examples

For example, given the following request header:

```
CH: dh=598, dw=384, dpr=2.0
```

The server knows that the client's screen height is 598px, width is 384px, as measured by density independent pixels on the device, and that the device pixel ratio is 2.0.

[2.4.](#) Server opt-in with Hop and Origin Hints

CH is an optional header which may be sent by the client when making a request to the server. The client may decide to always send the header, or use an optional opt-in mechanism, such as a predefined list of origins, user specified list of origins, or any other forms of opt-in.

For example, the server may advertise its support for Client Hints via Hop and/or Origin Hint ([\[I-D.nottingham-http-browser-hints\]](#)):

```
OH: ch
```

When a client receives the Hop or Origin Hint header indicating support for Client Hint adaptation, it should append the CH header to subsequent requests to the same origin server. Further, the client may remember this hint and automatically append the CH header for all future requests to the same origin.

2.5. Interaction with Caches

Client Hints may be combined with Key ([\[I-D.fielding-http-key\]](#)) to enable fine-grained control of the cache key for improved cache efficiency. For example, the server may return the following set of instructions:

```
Key: CH;pr=dw[320:640]
```

Above example indicates that the cache key should be based on the CH header, and the asset should be cached and made available for any client whose device width (dw) falls between 320 and 640 px.

```
Key: CH;pr=dpr[1.5:]
```

Above examples indicates that the cache key should be based on the CH header, and the asset should be cached and made available for any client whose device pixel ratio (dpr) is 1.5, or higher.

In absence of support for fine-grained control of the cache key via the Key header field, Vary response header can be used to indicate that served resource has been adapted based on specified Client Hint preferences.

```
Vary: CH
```

2.6. Relationship to the User-Agent Request Header

Client Hints does not supersede or replace User-Agent. Existing device detection mechanisms can continue to use both mechanisms if necessary. By advertising its capabilities within a request header, Client Hints allows for cache friendly and proactive content negotiation.

3. IANA Considerations

3.1. The CH Request Header Field

This document defines the "CH" HTTP request field, and registers it in the Permanent Message Headers registry.

- o Header field name: CH
- o Applicable protocol: HTTP
- o Status: Informational
- o Author/Change controller: Ilya Grigorik, ilya@igvita.com

- o Specification document(s): [this document]
- o Related information: for Client Hints

3.2. The HTTP Hints

This document registers the "ch" HTTP Hint ([[I-D.nottingham-http-browser-hints](#)]), as defined in [section 2.1](#):

- o Hint Name: ch
- o Hint Type: origin, hop
- o Description: When present, this hint indicates support for Client-Hints adaptation.
- o Value Type: numeric
- o Contact: ilya@igvita.com
- o Specification: this document

3.3. The HTTP Client Hints Registry

This document establishes the HTTP Client Hints Registry.

New hints are registered using Expert Review (see [[RFC5226](#)]), by sending e-mail to iana@iana.org (or using other mechanisms, as established by IANA).

New hints are expected to be implemented in at least one client in common use. The Expert MAY use their judgement in determining what "common" is, and when something is considered to be implemented.

New hints MUST be optional; they cannot place requirements upon implementations. Specifically, new hints MUST NOT make communication non-conformant with HTTP itself; i.e., this is not a mechanism for changing the HTTP protocol in incompatible ways.

See [section 2.1](#) for constraints on the syntax of hint names and hint values.

Registration requests MUST use the following template:

- o Hint Name: [name of hint]
- o Hint Value: ["boolean" or "numeric"]
- o Description: [description of hint]
- o Contact: [e-mail address(es)]
- o Specification: [optional; reference or URI to more info]
- o Notes: [optional]

The initial contents of the registry are defined in [section 2.2](#).

4. Security Considerations

The client controls which header fields are communicated and when. In cases such as incognito or anonymous profile browsing, the header can be omitted if necessary.

5. Normative References

[I-D.fielding-http-key]

Fielding, R. and M. Nottingham, "The Key HTTP Response Header Field", [draft-fielding-http-key-02](#) (work in progress), February 2013.

[I-D.ietf-httpbis-p1-messaging]

Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [draft-ietf-httpbis-p1-messaging-22](#) (work in progress), February 2013.

[I-D.ietf-httpbis-p2-semantic]

Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [draft-ietf-httpbis-p2-semantic-22](#) (work in progress), February 2013.

[I-D.nottingham-http-browser-hints]

Nottingham, M., "HTTP Origin and Hop Hints", [draft-nottingham-http-browser-hints-05](#) (work in progress), February 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.

Author's Address

Ilya Grigorik

Email: ilya@igvita.com

URI: <http://www.igvita.com/>