

**HTTP Client Hints**  
**draft-ietf-httpbis-client-hints-06**

Abstract

An increasing diversity of Web-connected devices and software capabilities has created a need to deliver optimized content for each device.

This specification defines an extensible and configurable set of HTTP request header fields, colloquially known as Client Hints, to address this. They are intended to be used as input to proactive content negotiation; just as the Accept header field allows user agents to indicate what formats they prefer, Client Hints allow user agents to indicate device and agent specific preferences.

Note to Readers

Discussion of this draft takes place on the HTTP working group mailing list ([ietf-http-wg@w3.org](mailto:ietf-http-wg@w3.org)), which is archived at <https://lists.w3.org/Archives/Public/ietf-http-wg/> [1].

Working Group information can be found at <http://httpwg.github.io/> [2]; source code and issues list for this draft can be found at <https://github.com/httpwg/http-extensions/labels/client-hints> [3].

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 January 17, 2019.

## Copyright Notice

Copyright (c) 2018 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 (<https://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

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">3</a>
<a href="#">1.1.</a>	<a href="#">Notational Conventions</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Client Hint Request Header Fields</a>	<a href="#">4</a>
<a href="#">2.1.</a>	<a href="#">Sending Client Hints</a>	<a href="#">4</a>
<a href="#">2.2.</a>	<a href="#">Server Processing of Client Hints</a>	<a href="#">4</a>
<a href="#">2.2.1.</a>	<a href="#">Advertising Support via Accept-CH Header Field</a>	<a href="#">5</a>
<a href="#">2.2.2.</a>	<a href="#">The Accept-CH-Lifetime Header Field</a>	<a href="#">5</a>
<a href="#">2.2.3.</a>	<a href="#">Interaction with Caches</a>	<a href="#">6</a>
<a href="#">3.</a>	<a href="#">Client Hints</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">The DPR Header Field</a>	<a href="#">6</a>
<a href="#">3.1.1.</a>	<a href="#">Confirming Selected DPR</a>	<a href="#">7</a>
<a href="#">3.2.</a>	<a href="#">The Width Header Field</a>	<a href="#">7</a>
<a href="#">3.3.</a>	<a href="#">The Viewport-Width Header Field</a>	<a href="#">7</a>
<a href="#">4.</a>	<a href="#">Examples</a>	<a href="#">8</a>
<a href="#">5.</a>	<a href="#">Security Considerations</a>	<a href="#">8</a>
<a href="#">6.</a>	<a href="#">IANA Considerations</a>	<a href="#">9</a>
<a href="#">6.1.</a>	<a href="#">Accept-CH</a>	<a href="#">9</a>
<a href="#">6.2.</a>	<a href="#">Accept-CH-Lifetime</a>	<a href="#">9</a>
<a href="#">6.3.</a>	<a href="#">Content-DPR</a>	<a href="#">10</a>
<a href="#">6.4.</a>	<a href="#">DPR</a>	<a href="#">10</a>
<a href="#">6.5.</a>	<a href="#">Viewport-Width</a>	<a href="#">10</a>
<a href="#">6.6.</a>	<a href="#">Width</a>	<a href="#">10</a>
<a href="#">7.</a>	<a href="#">References</a>	<a href="#">10</a>
<a href="#">7.1.</a>	<a href="#">Normative References</a>	<a href="#">10</a>
<a href="#">7.2.</a>	<a href="#">Informative References</a>	<a href="#">12</a>
<a href="#">7.3.</a>	<a href="#">URIs</a>	<a href="#">12</a>
<a href="#">Appendix A.</a>	<a href="#">Interaction with Key Response Header Field</a>	<a href="#">12</a>
<a href="#">Appendix B.</a>	<a href="#">Changes</a>	<a href="#">12</a>
<a href="#">B.1.</a>	<a href="#">Since -00</a>	<a href="#">12</a>
<a href="#">B.2.</a>	<a href="#">Since -01</a>	<a href="#">13</a>
<a href="#">B.3.</a>	<a href="#">Since -02</a>	<a href="#">13</a>

Grigorik

Expires January 17, 2019

[Page 2]

<a href="#">B.4.</a>	Since -03 . . . . .	<a href="#">13</a>
<a href="#">B.5.</a>	Since -04 . . . . .	<a href="#">13</a>
<a href="#">B.6.</a>	Since -05 . . . . .	<a href="#">13</a>
<a href="#">B.7.</a>	Since -06 . . . . .	<a href="#">13</a>
Acknowledgements	. . . . .	<a href="#">13</a>
Author's Address	. . . . .	<a href="#">13</a>

## **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 ([Section 5.5.3 of \[RFC7231\]](#)) header field 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, user agent sniffing has numerous limitations:

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

A popular alternative strategy is to use HTTP cookies ([\[RFC6265\]](#)) to communicate some information about the user agent. 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 set of new request header fields that allow user agent to perform proactive content negotiation ([Section 3.4.1 of \[RFC7231\]](#)) by indicating device and agent specific preferences, through a mechanism similar to the Accept header field which is used to indicate preferred response formats.

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



### **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.

This document uses the Augmented Backus-Naur Form (ABNF) notation of [[RFC5234](#)] with the list rule extension defined in [[RFC7230](#)], [Appendix B](#). It includes by reference the DIGIT rule from [[RFC5234](#)] and the OWS and field-name rules from [[RFC7230](#)].

## **2. Client Hint Request Header Fields**

A Client Hint request header field is a HTTP header field that is used by HTTP clients to indicate configuration data that can be used by the server to select an appropriate response. Each one conveys client preferences that the server can use to adapt and optimize the response.

### **2.1. Sending Client Hints**

Clients control which Client Hints are sent in requests, based on their default settings, user configuration, and server preferences. The client and server can use an opt-in mechanism outlined below to negotiate which fields should be sent to allow for efficient content adaption, and optionally use additional mechanisms to negotiate delegation policies that control access of third parties to same fields.

Implementers should be aware of the passive fingerprinting implications when implementing support for Client Hints, and follow the considerations outlined in "Security Considerations" section of this document.

### **2.2. Server Processing of Client Hints**

When presented with a request that contains one or more client hint header fields, servers can optimize the response based upon the information in them. When doing so, and if the resource is cacheable, the server MUST also generate a Vary response header field ([Section 7.1.4 of \[RFC7231\]](#)) to indicate which hints can affect the selected response and whether the selected response is appropriate for a later request.

Further, depending on the hint used, the server can generate additional response header fields to convey related values to aid



client processing. For example, this specification defines the "Content-DPR" response header field that needs to be returned by the server when the "DPR" hint is used to select the response.

### **2.2.1. Advertising Support via Accept-CH Header Field**

Servers can advertise support for Client Hints using the Accept-CH header field or an equivalent HTML meta element with http-equiv attribute ([HTML5]).

Accept-CH = #field-name

For example:

Accept-CH: DPR, Width, Viewport-Width

When a client receives an HTTP response advertising support for Client Hints, it should process it as origin ([RFC6454]) opt-in to receive Client Hint header fields advertised in the field-value. The opt-in MUST be delivered over a secure transport.

For example, based on Accept-CH example above, a user agent could append DPR, Width, and Viewport-Width header fields to all same-origin resource requests initiated by the page constructed from the response.

### **2.2.2. The Accept-CH-Lifetime Header Field**

Servers can ask the client to remember the set of Client Hints that the server supports for a specified period of time, to enable delivery of Client Hints on subsequent requests to the server's origin ([RFC6454]).

Accept-CH-Lifetime = #delta-seconds

When a client receives an HTTP response that contains Accept-CH-Lifetime header field, the field-value indicates that the Accept-CH preference SHOULD be persisted and bound to the origin, and be considered stale after response's age ([RFC7234], section 4.2) is greater than the specified number of seconds. The preference MUST be delivered over a secure transport, and MUST NOT be persisted for an origin that isn't HTTPS.

Accept-CH: DPR, Width  
Accept-CH: Viewport-Width  
Accept-CH-Lifetime: 86400





For example, based on the Accept-CH and Accept-CH-Lifetime example above, which is received in response to a user agent navigating to "https://example.com", and delivered over a secure transport: a user agent SHOULD persist an Accept-CH preference bound to "https://example.com" for up to 86400 seconds (1 day), and use it for user agent navigations to "https://example.com" and any same-origin resource requests initiated by the page constructed from the navigation's response. This preference SHOULD NOT extend to resource requests initiated to "https://example.com" from other origins.

If Accept-CH-Lifetime occurs in a message more than once, the last value overrides all previous occurrences.

### **2.2.3. Interaction with Caches**

When selecting an optimized response based on one or more Client Hints, and if the resource is cacheable, the server needs to generate a Vary response header field ([RFC7234]) to indicate which hints can affect the selected response and whether the selected response is appropriate for a later request.

Vary: DPR

Above example indicates that the cache key needs to include the DPR header field.

Vary: DPR, Width

Above example indicates that the cache key needs to include the DPR and Width header fields.

## **3. Client Hints**

### **3.1. The DPR Header Field**

The "DPR" request header field is a number that indicates the client's current Device Pixel Ratio (DPR), which is the ratio of physical pixels over CSS px (Section 5.2 of [CSSVAL]) of the layout viewport (Section 9.1.1 of [CSS2]) on the device.

DPR = 1\*DIGIT [ "." 1\*DIGIT ]

If DPR occurs in a message more than once, the last value overrides all previous occurrences.



### **3.1.1. Confirming Selected DPR**

The "Content-DPR" response header field is a number that indicates the ratio between physical pixels over CSS px of the selected image response.

Content-DPR = 1\*DIGIT [ "." 1\*DIGIT ]

DPR ratio affects the calculation of intrinsic size of image resources on the client - i.e. typically, the client automatically scales the natural size of the image by the DPR ratio to derive its display dimensions. As a result, the server **MUST** explicitly indicate the DPR of the selected image response whenever the DPR hint is used, and the client **MUST** use the DPR value returned by the server to perform its calculations. In case the server returned Content-DPR value contradicts previous client-side DPR indication, the server returned value **MUST** take precedence.

Note that DPR confirmation is only required for image responses, and the server does not need to confirm the resource width as this value can be derived from the resource itself once it is decoded by the client.

If Content-DPR occurs in a message more than once, the last value overrides all previous occurrences.

### **3.2. The Width Header Field**

The "Width" request header field is a number that indicates the desired resource width in physical px (i.e. intrinsic size of an image). The provided physical px value is a number rounded to the smallest following integer (i.e. ceiling value).

Width = 1\*DIGIT

If the desired resource width is not known at the time of the request or the resource does not have a display width, the Width header field can be omitted. If Width occurs in a message more than once, the last value overrides all previous occurrences.

### **3.3. The Viewport-Width Header Field**

The "Viewport-Width" request header field is a number that indicates the layout viewport width in CSS px. The provided CSS px value is a number rounded to the smallest following integer (i.e. ceiling value).

Viewport-Width = 1\*DIGIT



If Viewport-Width occurs in a message more than once, the last value overrides all previous occurrences.

#### 4. Examples

For example, given the following request header fields:

```
DPR: 2.0
Width: 320
Viewport-Width: 320
```

The server knows that the device pixel ratio is 2.0, that the intended display width of the requested resource is 160 CSS px (320 physical pixels at 2x resolution), and that the viewport width is 320 CSS px.

If the server uses above hints to perform resource selection for an image asset, it must confirm its selection via the Content-DPR response header to allow the client to calculate the appropriate intrinsic size of the image response. The server does not need to confirm resource width, only the ratio between physical pixels and CSS px of the selected image resource:

```
Content-DPR: 1.0
```

The Content-DPR response header field indicates to the client that the server has selected resource with DPR ratio of 1.0. The client can use this information to perform additional processing on the resource - for example, calculate the appropriate intrinsic size of the image resource such that it is displayed at the correct resolution.

#### 5. Security Considerations

The request header fields defined in this specification, and those that extend it, expose information about the user's environment to enable proactive content negotiation. Such information may reveal new information about the user and implementers ought to consider the following considerations, recommendations, and best practices.

Transmitted Client Hints header fields SHOULD NOT provide new information that is otherwise not available to the application via other means, such as using HTML, CSS, or JavaScript. Further, sending highly granular data, such as image and viewport width may help identify users across multiple requests. Reducing the set of field values that can be expressed, or restricting them to an enumerated range where the advertised value is close but is not an exact representation of the current value, can improve privacy and



reduce risk of linkability by ensuring that the same value is sent by multiple users. However, such precautions can still be insufficient for some types of data, especially data that can change over time.

Implementers ought to consider both user and server controlled mechanisms and policies to control which Client Hints header fields are advertised:

- o Implementers SHOULD restrict delivery of some or all Client Hints header fields to the opt-in origin only, unless the opt-in origin has explicitly delegated permission to another origin to request Client Hints header fields.
- o Implementers MAY provide user choice mechanisms so that users may balance privacy concerns with bandwidth limitations. However, implementers should also be aware that explaining the privacy implications of passive fingerprinting to users may be challenging.
- o Implementations specific to certain use cases or threat models MAY avoid transmitting some or all of Client Hints header fields. For example, avoid transmission of header fields that can carry higher risks of linkability.

Implementers SHOULD support Client Hints opt-in mechanisms and MUST clear persisted opt-in preferences when site data, browsing history, browsing cache, or similar, are cleared.

## **6. IANA Considerations**

This document defines the "Accept-CH", "DPR", "Viewport-Width", and "Width" HTTP request fields, "Accept-CH", "Accept-CH-Lifetime", and "Content-DPR" HTTP response field, and registers them in the Permanent Message Header Fields registry.

### **6.1. Accept-CH**

- o Header field name: Accept-CH
- o Applicable protocol: HTTP
- o Status: standard
- o Author/Change controller: IETF
- o Specification document(s): [Section 2.2.1](#) of this document
- o Related information: for Client Hints

### **6.2. Accept-CH-Lifetime**

- o Header field name: Accept-CH-Lifetime
- o Applicable protocol: HTTP
- o Status: standard
- o Author/Change controller: IETF





- o Specification document(s): [Section 2.2.2](#) of this document
- o Related information: for Client Hints

### **6.3. Content-DPR**

- o Header field name: Content-DPR
- o Applicable protocol: HTTP
- o Status: standard
- o Author/Change controller: IETF
- o Specification document(s): [Section 3.1.1](#) of this document
- o Related information: for Client Hints

### **6.4. DPR**

- o Header field name: DPR
- o Applicable protocol: HTTP
- o Status: standard
- o Author/Change controller: IETF
- o Specification document(s): [Section 3.1](#) of this document
- o Related information: for Client Hints

### **6.5. Viewport-Width**

- o Header field name: Viewport-Width
- o Applicable protocol: HTTP
- o Status: standard
- o Author/Change controller: IETF
- o Specification document(s): [Section 3.3](#) of this document
- o Related information: for Client Hints

### **6.6. Width**

- o Header field name: Width
- o Applicable protocol: HTTP
- o Status: standard
- o Author/Change controller: IETF
- o Specification document(s): [Section 3.2](#) of this document
- o Related information: for Client Hints

## **7. References**

### **7.1. Normative References**

- [CSS2] Bos, B., Celic, T., Hickson, I., and H. Lie, "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", W3C Recommendation REC-CSS2-20110607, June 2011, <<http://www.w3.org/TR/2011/REC-CSS2-20110607>>.



- [CSSVAL] Atkins, T. and E. Etemad, "CSS Values and Units Module Level 3", World Wide Web Consortium CR CR-css-values-3-20160929, September 2016, <<https://www.w3.org/TR/2016/CR-css-values-3-20160929>>.
- [HTML5] Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, T., and S. Pfeiffer, "HTML5", World Wide Web Consortium Recommendation REC-html5-20141028, October 2014, <<http://www.w3.org/TR/2014/REC-html5-20141028>>.
- [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>>.
- [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>>.
- [RFC6454] Barth, A., "The Web Origin Concept", [RFC 6454](#), DOI 10.17487/RFC6454, December 2011, <<https://www.rfc-editor.org/info/rfc6454>>.
- [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>>.
- [RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", [RFC 7234](#), DOI 10.17487/RFC7234, June 2014, <<https://www.rfc-editor.org/info/rfc7234>>.
- [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>>.



## **7.2. Informative References**

- [KEY] Fielding, R. and M. Nottingham, "The Key HTTP Response Header Field", [draft-ietf-httpbis-key-01](#) (work in progress), March 2016.
- [RFC6265] Barth, A., "HTTP State Management Mechanism", [RFC 6265](#), DOI 10.17487/RFC6265, April 2011, <<https://www.rfc-editor.org/info/rfc6265>>.

## **7.3. URIs**

- [1] <https://lists.w3.org/Archives/Public/ietf-http-wg/>
- [2] <http://httpwg.github.io/>
- [3] <https://github.com/httpwg/http-extensions/labels/client-hints>

## **Appendix A. Interaction with Key Response Header Field**

Client Hints may be combined with Key response header field ([[KEY](#)]) to enable fine-grained control of the cache key for improved cache efficiency. For example, the server can return the following set of instructions:

Key: DPR;partition=1.5:2.5:4.0

Above example indicates that the cache key needs to include the value of the DPR header field with three segments: less than 1.5, 1.5 to less than 2.5, and 4.0 or greater.

Key: Width;div=320

Above example indicates that the cache key needs to include the value of the Width header field and be partitioned into groups of 320: 0-320, 320-640, and so on.

## **Appendix B. Changes**

### **B.1. Since -00**

- o Issue 168 (make Save-Data extensible) updated ABNF.
- o Issue 163 (CH review feedback) editorial feedback from httpwg list.
- o Issue 153 (NetInfo API citation) added normative reference.



**B.2. Since -01**

- o Issue 200: Moved Key reference to informative.
- o Issue 215: Extended passive fingerprinting and mitigation considerations.
- o Changed document status to experimental.

**B.3. Since -02**

- o Issue 239: Updated reference to CR-css-values-3
- o Issue 240: Updated reference for Network Information API
- o Issue 241: Consistency in IANA considerations
- o Issue 250: Clarified Accept-CH

**B.4. Since -03**

- o Issue 284: Extended guidance for Accept-CH
- o Issue 308: Editorial cleanup
- o Issue 306: Define Accept-CH-Lifetime

**B.5. Since -04**

- o Issue 361: Removed Downlink
- o Issue 361: Moved Key to appendix, plus other editorial feedback

**B.6. Since -05**

- o Issue 372: Scoped CH opt-in and delivery to secure transports
- o Issue 373: Bind CH opt-in to origin

**B.7. Since -06**

- o Issue 524: Save-Data is now defined by NetInfo spec, dropping

**Acknowledgements**

Thanks to Mark Nottingham, Julian Reschke, Chris Bentzel, Yoav Weiss, Ben Greenstein, Tarun Bansal, Roy Fielding, Vasiliy Faronov, Ted Hardie, Jonas Sicking, and numerous other members of the IETF HTTP Working Group for invaluable help and feedback.

**Author's Address**

Ilya Grigorik  
Google

Email: [ilya@igvita.com](mailto:ilya@igvita.com)

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



