HTTP Client hints

- Overview
- Caching and Client Hints
- Hint life-time
- Low entropy hints
- High entropy hints
- Critical client hints
- Hint types
  - User agent client hints
  - User preference media features client hints
  - Device client hints
  - Network client hints
- Using client hints for responsive design
- See also

Client hints are a set of HTTP request header fields that a server can proactively request from a client to get information about the device, network, user, and user-agent-specific preferences. The server can determine which resources to send, based on the information that the client chooses to provide.

The set of "hint" headers are listed in the topic HTTP Headers and summarized below.

Overview

1. When the browser first makes a request to load a webpage, it will send the User-Agent header to the server.
2. Additionally, it will send the server a default set of `Sec-CH-UA-*` headers; this set of hints are referred to as the low entropy hints. An Android device, for example, would send something like this:

```
Sec-CH-UA: "Google Chrome";v="143", "Chromium";v="143", "Not A(Brand";v="24"
Sec-CH-UA-Platform: "Android"
Sec-CH-UA-Mobile: ?1
```

These headers provide the following information:

- Sec-CH-UA: The major browser version and other brands associated with it.
- Sec-CH-UA-Platform: The platform.
- Sec-CH-UA-Mobile: A boolean that indicates whether the browser is running on a mobile device (`?1`) or not (`?0`).

3. The server can announce that it supports client hints and request additional client hints using the Accept-CH response header, which contains a comma-delimited list of the additional headers it would like to receive in subsequent requests. For example:

   ```
   Accept-CH: Sec-CH-UA-Model, Sec-CH-UA-Form-Factors
   ```

   The default set of headers is always sent; in this case, we've also requested:

   - Sec-CH-UA-Model: The device model the platform is running on.
   - Sec-CH-UA-Form-Factors: The device's form factor(s), which indicate how the user interacts with the user-agent — the screen size, controls, etc.

4. If the browser is permitted, it will send the requested headers in all subsequent requests, until the browser or tab is closed. For example, our example Android phone might send the following updated headers with subsequent requests:

   ```
   Sec-CH-UA: "Google Chrome";v="143", "Chromium";v="143", "Not A(Brand";v="24"
   Sec-CH-UA-Platform: "Android"
   Sec-CH-UA-Mobile: ?1
   Sec-CH-UA-Model: "Pixel 9"
   Sec-CH-UA-Form-Factors: "Mobile"
   ```

This approach is efficient in that the server only requests the information that it is able to usefully handle. It is also relatively "privacy-preserving", in that it is up to the client to decide what information it can safely share.

> **NOTE:** Client hints can also be specified in HTML using the meta element with the `http-equiv` attribute.
>
> ```
> <meta http-equiv="Accept-CH" content="Width, Downlink, Sec-CH-UA" />
> ```

Caching and Client Hints

Client hints that determine which resources are sent in responses should generally also be included in the affected response's Vary header. This ensures that a different resource is cached for every different value of the hint header.

```
Vary: Accept, Width, ECT
```

You may prefer to omit specifying Vary or use some other strategy for client hint headers where the value changes a lot, as this effectively makes the resource uncacheable. (A new cache entry is created for every unique value.) This applies in particular to network client hints like Downlink and RTT. For more information see HTTP Caching > Vary.

Hint life-time

A server specifies the client hint headers that it is interested in getting in the `Accept-CH` response header. The user agent appends the requested client hint headers, or at least the subset that it wants to share with that server, to all subsequent requests in the current browsing session.

In other words, the request for a specific set of hints does not expire until the browser is shut down.

A server can replace the set of client hints it is interested in receiving by resending the `Accept-CH` response header with a new list. For example, to stop requesting any hints it would send `Accept-CH` with an empty list.

> **NOTE:** The client hints set for a particular origin can also be cleared by sending a Clear-Site-Data: "clientHints" response header for a URL inside that origin.

Low entropy hints

Client hints are broadly divided into high and low entropy hints. The low entropy hints are those that don't give away much information that might be used to fingerprint a user. They may be sent by default on every client request, irrespective of the server `Accept-CH` response header, depending on the permission policy. Low entropy hints are:

- Save-Data,
- Sec-CH-UA,
- Sec-CH-UA-Mobile, and
- Sec-CH-UA-Platform.

High entropy hints

The high entropy hints are those that have the potential to give away more information that can be used for user fingerprinting, and therefore are gated in such a way that the user agent can make a decision whether to provide them. The decision might be based on user preferences, a permission request, or a permission policy. All client hints that are not low entropy hints are considered high entropy hints.

Critical client hints

A *critical client hint* is one where applying the response may significantly change the rendered page, potentially in a way that is jarring or will affect usability, and therefore which must be applied before the content is rendered. For example, `Sec-CH-Prefers-Reduced-Motion` is commonly treated as a critical hint, because it might markedly affect the behavior of animations, and because a user who chooses this preference needs it to be set.

A server can use the Critical-CH response header along with `Accept-CH` to specify that an accepted client hint is also a critical client hint (a header in `Critical-CH` must also appear in `Accept-CH`). User agents receiving a response with `Critical-CH` must check if the indicated critical headers were sent in the original request. If not, then the user agent will retry the request rather than render the page. This approach ensures that client preferences set using critical client hints are always used, even if not included in the first request, or if the server configuration changes.

For example, in this case, the server tells a client via Accept-CH that it accepts `Sec-CH-Prefers-Reduced-Motion`, and Critical-CH is used to specify that `Sec-CH-Prefers-Reduced-Motion` is considered a critical client hint:

```
HTTP/1.1 200 OK
Content-Type: text/html
Accept-CH: Sec-CH-Prefers-Reduced-Motion
Vary: Sec-CH-Prefers-Reduced-Motion
Critical-CH: Sec-CH-Prefers-Reduced-Motion
```

> **NOTE:** We've also specified `Sec-CH-Prefers-Reduced-Motion` in the Vary header to indicate to the browser that the served content will differ based on this header value, even if the URL stays the same, so the browser shouldn't just use an existing cached response and instead should cache this response separately. Each header listed in the `Critical-CH` header should also be present in the `Accept-CH` and `Vary` headers.

As `Sec-CH-Prefers-Reduced-Motion` is a critical hint that was not in the original request, the client automatically retries the request — this time telling the server via `Sec-CH-Prefers-Reduced-Motion` that it has a user preference for reduced-motion animations.

```
GET / HTTP/1.1
Host: example.com
Sec-CH-Prefers-Reduced-Motion: "reduce"
```

In summary, `Accept-CH` requests all values you'd like for the page, while `Critical-CH` requests only the subset of values you must have on-load to properly load the page.

Hint types

User agent client hints

User agent (UA) client hint headers allow a server to vary responses based on the user agent (browser), operating system, and device. For a list of `Sec-CH-UA-*` headers, see User agent client hints headers.

Client hints are available to web page JavaScript via the User Agent Client Hints API.

> **NOTE:** Servers currently get most of the same information by parsing the User-Agent header. For historical reasons this header contains a lot of largely irrelevant information, and information that might be used to identify a *particular user*. UA client hints provide a more efficient and privacy preserving way of getting the desired information. They are eventually expected to replace this older approach.

> **NOTE:** User-agent client hints are not available inside fenced frames because they rely on permissions policy delegation, which could be used to leak data.

User preference media features client hints

User Preference Media Features Client Hints allow a server to vary responses based on a user agent's preferences for CSS media features such as color scheme or reduced motion. Headers include: Sec-CH-Prefers-Reduced-Motion, Sec-CH-Prefers-Color-Scheme.

Device client hints

Device client hints allow a server to vary responses based on device characteristics including available memory and screen properties. Headers include: Sec-CH-Device-Memory, Sec-CH-DPR, Sec-CH-Viewport-Height, Sec-CH-Viewport-Width.

Network client hints

Network client hints allow a server to vary responses based on the user's choice, network bandwidth, and latency. Headers include: Save-Data, Downlink, ECT, RTT.

Using client hints for responsive design

It's possible to use client hints for responsive design, for example to detect whether a mobile device or tablet is rendering your site.

An Android phone would send the following default client hints:

```
Sec-CH-UA: "Google Chrome";v="143", "Chromium";v="143", "Not A(Brand";v="24"
Sec-CH-UA-Platform: "Android"
Sec-CH-UA-Mobile: ?1
```

An Android tablet on the other hand would send the following:

```
Sec-CH-UA: "Google Chrome";v="143", "Chromium";v="143", "Not A(Brand";v="24"
Sec-CH-UA-Platform: "Android"
Sec-CH-UA-Mobile: ?0
```

The Sec-CH-UA-Mobile header can be used to determine whether the device is a mobile device or not. For hardware-specific use cases, the device model name and form factor can be requested via the high-entropy Sec-CH-UA-Model and Sec-CH-UA-Form-Factors hints.

For many responsive design needs, media queries may be more convenient and flexible. However, there may be cases where you don't have control over the individual stylesheets of a site, and need to vary the stylesheet served based on the device signature or some kind of user preference. There are client hints that mirror some of the "user preference" media queries, such as Sec-CH-Prefers-Color-Scheme, Sec-CH-Prefers-Reduced-Motion, and Sec-CH-Prefers-Reduced-Transparency.

See also

- Client Hints headers
- `Vary` HTTP Header
- Client Hints Infrastructure
- User Agent Client Hints API
- Improving user privacy and developer experience with User-Agent Client Hints on developer.chrome.com (2020)
- Migrate to User-Agent Client Hints on web.dev (2021)

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:35:27 UTC