HTTP No-Vary-Search

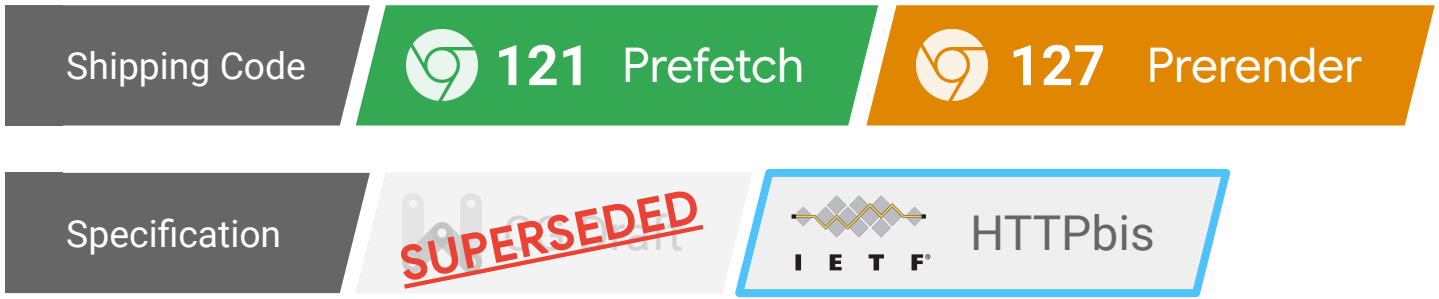
Jeremy Roman, Google Chrome

IETF 121 – HTTPbis

What does this look like?

- `No-Vary-Search: params`
  This response does not depend on query params.
- `No-Vary-Search: key-order`
  This response does not depend on the relative order of different params.
- `No-Vary-Search: params=("utm_source")`
  This response does not depend on the utm_source param.
- `No-Vary-Search: key-order, params, except=("productId" "size")`
  This response does not depend on the order of params, nor the values of params except productId and size.

Specification & Implementation Status



Integration

If a cache [HTTP-CACHING] implements this specification, the presented target URI requirement in Section 4 of [HTTP-CACHING] is replaced with:

- one of the following:
  - the presented target URI (Section 7.1 of [HTTP]) and that of the stored response match, or
  - the presented target URI and that of the stored response are equivalent modulo search variance (Section 5), given the variance obtained (Section 4.2) from the stored response.

Efficient Implementation

Servers **SHOULD** send no more than one distinct non-empty value for the `No-Vary-Search` field in response to requests for a given pathname.

Cache implementations **MAY** fail to reuse a stored response whose target URI matches *only* modulo URL search variance, if the cache has more recently stored a response which:

- has a target URI which is equal to the presented target URI, excluding the query, and
- has a non-empty value for the `No-Vary-Search` field, and
- has a `No-Vary-Search` field value different from the stored response being considered for reuse.

Internet-Draft
https://httpwg.org/http-extensions/draft-ietf-httpbis-no-vary-search.html

Issues
https://github.com/httpwg/http-extensions/labels/no-vary-search

Backup Slides

In case a question about these topics is asked

Why?

- In practice, the URL search component is treated as a **list of key-value pairs**, at least on the web.
- Just as responses do not depend on all **request headers**, they do not depend on all **URL query params**.
- **Caches work best** when they know as much as possible about which cached responses can be used to satisfy a request.
- Existing cache software supports ignoring query parameters, but HTTP does not provide a **cross-vendor way of indicating this** (whereas it does for request headers, in the form of `Vary`).
- Today, Chrome supports this for navigational prefetch and prerender. However, it seems **generally useful** to HTTP cache implementations (e.g., browser, proxies, CDNs).

Why do clients send "meaningless" parameters?

- URLs may have query parameters in a **different order** because the order is not significant (e.g., `a=1&b=2` and `b=2&a=1` have the same meaning).
- Parameters may **affect server processing but not the semantic meaning** of the result (e.g., load balancing to a particular backend instance, enabling debug logging, changing request priority).
- Parameters may carry data **intended for processing by client software** (e.g., JavaScript analytics code, initialization parameters for script on a web page) which does not affect the response's cache suitability.