

HTTP No-Vary-Search

Jeremy Roman, Google Chrome IETF 120 – HTTPbis

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).

Google

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.

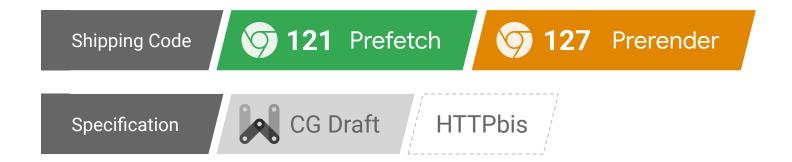
What does this look like?

- → 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



Outstanding specification work

- → any further **reformatting/restructuring** for the change in venue
- explicit integration with other IETF standards (notably RFC 9111 "HTTP Caching") for implementers of caches which implement those standards
 - → including how implementations may handle multiple matching responses
- addressing any issues we did not encounter in Chrome's prefetch cache but which other parties anticipate
- any extensions required to express semantics which are not yet captured but important to this use case

Explainer

https://github.com/WICG/nav-speculation/blob/main/no-vary-search.md

Internet-Draft

https://datatracker.ietf.org/doc/draft-wicg-http-no-vary-search/01/

WICG Draft

https://wicg.github.io/nav-speculation/no-vary-search.html