Engineer

Canonical URL

the one address a page says is its own

What it is

A page that answers at several addresses can name one of them as its original, and the address it names is its canonical URL: the copy it asks search engines and other consumers to index, display and credit. The name comes from the markup that carries it, a `link` element with `rel="canonical"`, so the same thing also goes by canonical link, canonical tag or simply rel=canonical.

```
<link rel="canonical" href="https://example.com/guide/" />
```

Duplicates arise more often than anyone plans for. One article can be reached with and without `www`, over `http` and `https`, with a tracking parameter on the end, sorted a different way, as a print view, or republished on a second host. RFC 6596, published in 2012, registered the relation and set its one hard rule: the target must identify content that is either a duplicate of the referring page or a superset of it. Within that rule the target may be the page itself, sit on another hostname, or use another scheme. The HTML standard describes it as the “preferred URL for the current document”. A file that has no HTML head, such as a PDF, can send the same statement as an HTTP `Link` header with `rel="canonical"`.

A canonical is a hint rather than a command. The RFC lets applications ignore a designation that is improper, and Google’s documentation ranks redirects and rel=canonical as strong signals and inclusion in a sitemap as a weak one. Where a page declares nothing, the search engine picks a copy itself.

It sits beside three tools that are easy to mix up with it. A permanent redirect removes the duplicate for people as well as machines, which makes it the stronger choice whenever the extra address has no reason to stay reachable. A

`noindex` directive keeps a page out of the index altogether, which is a different outcome from merging it into another page. And `hreflang` ties together pages that are translations of each other; translations are equivalents, and none of them is a duplicate of another.

Use the concept in a decision

Put “What is a canonical URL?” to work in a real decision. We can help relate the concept to the system you are building or buying. A site or an app that the AI assistants your customers use can find, read and act on, checked on every release rather than promised once.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Built a multilingual Hugo site in 3 languages — 108 template files serving three languages and four formats

Built a multilingual Hugo site in 3 languages Get in touch →

Why it matters

The decision a canonical makes is which address becomes the indexed one, and the only real choice is between making it deliberately and having it made by somebody else. Search engines do not punish duplicate text so much as collapse it: they keep one copy and fold the rest into it. The copy they keep is the one people see in results, the one that links to every variant get credited to, and the one a crawler returns to. Left to the engine, that copy can be the version with a campaign parameter, the staging host that leaked, or a partner’s republished copy instead of the original.

A redirect versus a canonical is mostly a question about people. If nobody needs the second address, redirect it permanently and the problem disappears for every client at once; RFC 6596 itself asks authors to rule out a permanent redirect before reaching for the link. If the copy has to stay reachable, because a campaign link points at it, a print layout lives there, or a second site carries the article for its own audience, then the canonical is the tool, because it settles the question for indexes and leaves readers alone.

The cost lies in consistency rather than in the tag. One line in the head is cheap. Every other place that names the page’s address has to agree with it: the `og:url` used by link previews, the identifiers in structured data, the sitemap, and the site’s own internal links. Two of those naming one address and the third naming another is a page making two claims about itself, and a consumer may believe either. What a canonical decides is which address any standing the content earns is attached to.

What it looks like in practice

This site is a demanding case for it. The same content tree publishes pages in several representations. The tree is also built for the focused subdomains.

The templates settle the duplicates with one rule: a page is its own original only if it was written for the address it is served from. On the main site that covers everything, so every page names itself. A focused subdomain wrote its own front page and nothing else, so that page names itself while the mirrored work, the shared pages and every list the generator builds point at the main site's copy at the same path. The default runs in the safe direction: everything points home unless it says otherwise, and a page that deserves its own identity has to ask for it in front matter, where the asking is visible. A new kind of page therefore inherits the conservative answer instead of quietly claiming to be an original.

The value is computed in one template and read everywhere it appears: by `rel="canonical"`, by `og:url` and `twitter:url`, and by the `@id` and `url` of the page's structured-data node. The sitemap and the language alternates ask the same question in yes-or-no form and are emitted only when the canonical is the page's own address, because requesting that one address be indexed while naming another as the original is a contradiction a crawler can resolve by trusting neither. A copy on a focused host still renders in full, and its language switcher still works; what it withholds is the request to be indexed, never the content. The Markdown twin is announced from the HTML page with `rel="alternate"` and its media type, which the HTML standard reads as the same document in another format rather than a rival page.

The main-site output check requires one absolute, self-referencing canonical on normal HTML pages and checks internal page links for missing trailing slashes. Standalone delivered documents and the social card follow their own policies; focused-host checks verify the distinction between mirrors and originals.

Where it goes wrong

Search results show an address nobody chose. A tracking parameter, a stray host or a mirror turns up as the listed URL. The page either declares no canonical, or declares one while its sitemap entry, `og:url` and internal links name something else, so the engine weighs mixed signals and picks. Emit one absolute canonical on every page, including a self-referencing one on the original, and derive the sitemap, preview and structured-data addresses from that same value.

Page two of a series drops out of the index. Every page of a paginated list or a split article names page one as its canonical. Page one is neither a duplicate nor a superset of page two, so whatever appears only on later pages can be treated as duplicate and disregarded, which RFC 6596 warns about by name. Let each page in a series name itself, or point every page at a single view-all page that really contains all of it.

A duplicate blocked in robots.txt stays listed, or the original vanishes with it. Blocking a duplicate's address in robots.txt stops the crawler from reading the page, including its canonical, and Google notes that a blocked URL can still be indexed without its content. Marking duplicates `noindex` removes them from search entirely instead of merging what they earned into the original. Leave duplicates crawlable and let a canonical or a redirect do the consolidating.

The declared canonical is ignored. The target redirects permanently somewhere else, names a different canonical of its own, returns a 404, or competes with a second canonical element on the same page; script can also rewrite the link after the HTML arrived. Each of these makes the designation improper, and the RFC lets a consumer fall back on its own judgement. Point at the final address that returns 200, emit one element in the served HTML head, and let no script change it afterwards.

Further reading

- RFC 6596: The Canonical Link Relation — the registration, the duplicate-or-superset rule and the designations to avoid.
- HTML Standard: link type "canonical" — how HTML defines the keyword and where it may appear.
- RFC 8288: Web Linking — the HTTP `Link` header that carries a canonical for files without an HTML head.
- Google Search Central: consolidate duplicate URLs — how one search engine weighs redirects, canonicals and sitemaps against each other.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is a canonical URL?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/canonical/