Engineer

Sitemap

a list of a site's pages, written for machines

What it is

A sitemap is a file a website publishes about itself: the addresses its owner wants search engines to know, one entry per page, each with a few optional facts attached. It exists because a crawler otherwise learns a site the slow way, by following links from pages it already knows, and some pages sit at the end of a long chain of links or have hardly any inbound links at all.

The format the major engines share is the Sitemaps protocol, published at sitemaps.org. Its XML version wraps the list in a `<urlset>` element with one `<url>` per page. Inside it, `<loc>` holds the full address and is the only required child. `<lastmod>` is the date the page last changed, `<changefreq>` a guess at how often it will change, and `<priority>` a value from 0.0 to 1.0 that ranks the page against the site's own other pages, never against anyone else's. The protocol also accepts a plain text file with one address per line, or an RSS or Atom feed.

The protocol sets hard edges. One file holds at most 50,000 addresses and 50 MB uncompressed, may be gzipped, and must be UTF-8 with its values entity-escaped; each address stays under 2,048 characters. A larger site splits its list and publishes a sitemap index, a second kind of file that names up to 50,000 sitemaps on the same site. A sitemap may also list only addresses at or below its own location: a file at `/catalog/sitemap.xml` covers the `/catalog/` pages and stops at that folder's edge.

A search engine learns where the file lives in one of two ways. A `Sitemap:` line in robots.txt gives its full address, independently of any user-agent group, or the owner submits it through the engine's own tools, such as Google Search Console.

The same word also names a page of links for human visitors; when the difference matters, people say "XML sitemap" for the one written for programs.

Use the concept in a decision

Put “What is a sitemap?” 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

Google’s documentation is candid about the limits. A site whose pages are properly linked can usually be discovered without a sitemap, and Google draws the line for “small” at about 500 pages that should appear in results. It also says a sitemap “doesn’t guarantee that all the items in your sitemap will be crawled and indexed”, and that submitting one “is merely a hint”. The cases where it helps most are large sites, new sites with few links from elsewhere, and sites heavy with video, images or news.

Almost every site generator writes one for free, so the real decision is what the file claims, and every entry is a statement an engine can check against the page itself.

Which addresses go in? Google asks for the canonical address of each page. A list padded with redirects, duplicates or pages marked `noindex` asks for exactly what the rest of the site says to leave alone.

Which fields? Google says it ignores `<priority>` and `<changefreq>`. It uses `<lastmod>` only “if it’s consistently and verifiably (for example by comparing to the last modification of the page) accurate”, and the protocol says the date “must be set to the date the linked page was last modified, not when the sitemap is generated”. Google counts changes to a page’s main content, structured data or links as significant, and a new copyright year as trivial.

That is where the cost sits. A date from the file system is free and usually wrong, because deploys, exports and fresh checkouts touch files whose words stayed the same. A date from version history is honest, but the build then needs that history present and read in the right order. A date typed by hand is precise until someone forgets to change it.

What it looks like in practice

This site's robots.txt names one address, `https://engineer.company/sitemap.xml`, and that file is a sitemap index naming the sitemaps for the published language editions. Each page entry carries the address and a date, plus links to the same page in its other published translations. `<priority>` and `<changefreq>` are left out, since Google says it ignores both and the protocol makes neither of them required. When this article was written, the entry for the page on AI readiness read like this, its language links set aside:

```
<url>
  <loc>https://engineer.company/ai-ready/</loc>
  <lastmod>2026-10-15T14:25:16+02:00</lastmod>
</url>
```

The list is selective on purpose. The company's focused satellite sites republish work from here with a canonical pointing back, and the same template keeps those copies out of each satellite's sitemap: an entry asks an engine to index that exact address, and asking for a copy while naming another page as the original would contradict itself. Each host announces only its own pages.

The dates have a history of their own. At one point the sitemap told search engines that 172 of its 176 pages had changed at the same instant. The cause was ordinary: the date came from each file's modification time, and the content export rewrites every file on every run. The fix moved the source to version history, with a fallback chain that tries an explicit field, then the commit history, then the file. It held one trap worth knowing: the literal field name has to appear in that list, or the generator never reads it. Afterwards, a re-sync following a month of edits touched eight files of 186 instead of all of them, and the sitemap says something true.

That date now does work beyond the sitemap. As the page on being found by AI agents describes, every deploy announces what changed through IndexNow, a protocol that Bing and several other engines take part in. The deploy script decides what changed by comparing each page's `lastmod` with what it last sent, so a deploy that edits one page announces one page. Google is not on IndexNow's own list of participating engines, so it learns of the change when it next reads the sitemap. A guard in the build keeps the dates honest: it reads

every sitemap date against the page's history and refuses to publish when one is wrong. For people rather than programs there is also a readable index of every page.

Where it goes wrong

Every page claims to have changed today. The sitemap carries one date, or one per deploy, across hundreds of pages, and an engine reading it learns nothing about which work is new. The cause is a date taken from the file system or the build clock. Google uses `<lastmod>` only when it is consistently and verifiably accurate, which a shared timestamp visibly is not. Take the date from the content's own history, and compare a sample of entries against it after any change to the build.

The list asks for pages the site wants left out. Redirects, duplicate addresses carrying tracking parameters, pages marked `noindex`, staging hosts and copies whose canonical points elsewhere all appear in sitemaps written by a generator that lists every file it built. Each one sends a crawler to an address the page itself then disowns. List only the canonical address of each page meant for results, and let the sitemap and the canonical tag answer from one rule, as this site's templates do.

Entries fall outside what the file may cover. A sitemap at `/blog/sitemap.xml` lists shop pages, or writes relative paths such as `/page.html`. The protocol limits a file to addresses at or below its own folder, and Google crawls addresses exactly as written. Publish the sitemap at the root, write every `<loc>` as a complete `https://` address on the same host, and switch to a sitemap index before the list nears 50,000 entries or 50 MB.

Pages listed for months stay out of the index. The owner reads the sitemap as an instruction and waits. Google calls it a hint: a listed address can still be skipped or folded into a duplicate for reasons outside the file. Look at the page rather than the list: whether other pages link to it, whether its canonical names itself, and whether robots.txt or a `noindex` rule keeps a crawler away.

Further reading

- Sitemaps XML format — the protocol itself: every tag, the size limits, index files and which addresses a file may list.

- Build and submit a sitemap — Google's rules, including the tags it ignores and when it trusts a date.
- Learn about sitemaps — when Google says a site needs one and when good linking will do.
- IndexNow documentation — the protocol that tells participating engines about a change on the day it ships.
- RFC 9309: Robots Exclusion Protocol — the robots.txt standard, which lets a crawler read a `Sitemap:` line beside its rules.

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 sitemap?" 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/sitemap/