sitemap.xml: telling search engines what exists

Level: beginner · **Reading time:** ~8 minutes

`robots.txt` says where crawlers may **not** go. `sitemap.xml` says what is **there**. They are complements, not alternatives, and most sites need both.

A sitemap is a machine-readable list of every URL you want indexed, plus a hint about when each one last changed. That last part is what makes it valuable: without it, a crawler has to guess whether your pages are worth re-checking.

What it is for

Search engines discover pages by following links. That works, but it is slow and incomplete. A sitemap helps when:

- Your site is **new** and has few inbound links
- Pages are **deep** in the hierarchy, many clicks from the homepage
- Content changes **often** and you want re-crawls to be prompt
- You have **orphan pages** nothing links to

> A sitemap is a hint, not a command. Listing a URL does not guarantee indexing, and omitting one does not prevent it. Google decides what to index; the sitemap only makes discovery reliable.

What it looks like

The format is sitemaps.org XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://example.com/</loc>
    <lastmod>2026-08-21</lastmod>
    <changefreq>weekly</changefreq>
    <priority>1.0</priority>
  </url>
  <url>
    <loc>https://example.com/posts/hello-world/</loc>
    <lastmod>2026-08-14</lastmod>
```

```
  </url>
</urlset>
```

Tag	Required	Notes
`<loc>`	Yes	Absolute URL, correctly escaped
`<lastmod>`	No	W3C date. **Google uses this** — if it is accurate
`<changefreq>`	No	Google ignores it
`<priority>`	No	Google ignores it too

The practical takeaway: **spend your effort on accurate `lastmod` and ignore the other two.** Google has said plainly that it disregards `changefreq` and `priority`, and that it will stop trusting `lastmod` if it sees you setting it to the current date on every build regardless of real changes.

The limits

- **50,000 URLs** and **50 MB uncompressed** per file
- Over that, split into several files and publish a **sitemap index**
- Every URL must be on the same host as the sitemap (with exceptions if ownership of both is verified)

A sitemap index looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>https://example.com/sitemap-posts.xml</loc>
    <lastmod>2026-08-21</lastmod>
  </sitemap>
  <sitemap>
    <loc>https://example.com/sitemap-pages.xml</loc>
  </sitemap>
</sitemapindex>
```

Getting it found

Two steps, both worth doing:

1. **Reference it from robots.txt** — every major crawler reads this:

   ```
   Sitemap: https://example.com/sitemap.xml
   ```

2. **Submit it in Google Search Console** (Indexing → Sitemaps) and Bing Webmaster Tools. This also gives you an error report, which robots.txt alone does not.

> **The old "ping" URLs are dead.** `google.com/ping?sitemap=...` was retired in 2023.
> If a tutorial tells you to ping Google after each deploy, it is out of date.

For faster pickup on Bing and Yandex, IndexNow lets you push changed URLs instantly instead of waiting for a crawl. Google does not participate.

The mistakes

Listing URLs that redirect or 404. A sitemap full of dead links erodes trust in the whole file. Every `<loc>` should return `200`.

Listing non-canonical URLs. If `/page` canonicalises to `/page/`, list only the canonical form. Mixing them sends conflicting signals.

Including pages you block in robots.txt. A direct contradiction: "please index this / you may not read it."

Faking `lastmod`. Setting every URL to today's date on every deploy is the fastest way to make Google ignore the field permanently.

Forgetting trailing-slash consistency. `https://example.com/about` and `https://example.com/about/` are different URLs. Pick the one your site actually serves.

How to generate one

Without code

- XML-Sitemaps.com — crawls your live site, free up to 500 URLs
- Screaming Frog SEO Spider — desktop app, free up to 500 URLs, far more control
- **WordPress:** Yoast SEO, Rank Math, or WordPress core all generate one automatically. You almost never need a plugin specifically for this.
- **Shopify / Wix / Squarespace:** generated automatically at `/sitemap.xml`. Nothing to do.

Crawler-based generators have one real weakness: they only find pages that are already linked. An orphan page stays missing.

How to test it

1. `curl -s https://example.com/sitemap.xml | head -20` — confirm it serves `200` and `application/xml`
2. **Google Search Console → Sitemaps** — shows URLs discovered, indexed, and any parse errors
3. Spot-check ten random `<loc>` values for a `200` status
4. Validate the XML against the sitemaps.org schema

Checklist

- [ ] Reachable at `/sitemap.xml`, returns `200` and `application/xml`
- [ ] Referenced by an absolute `Sitemap:` line in `robots.txt`
- [ ] Submitted in Google Search Console and Bing Webmaster Tools
- [ ] Contains only canonical, `200`-returning URLs
- [ ] `lastmod` reflects real content changes, not build times
- [ ] Under 50,000 URLs and 50 MB, or split with an index
- [ ] Excludes anything blocked in `robots.txt`

References

- sitemaps.org protocol — the format itself
- Google: Learn about sitemaps
- Google: Build and submit a sitemap
- Google: sitemaps lastmod, changefreq, priority and ping
- Bing Webmaster Tools
- IndexNow
- Next.js sitemap.ts reference