# robots.txt: the first file every crawler asks for

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

Before Google, Bing, or ChatGPT's crawler reads a single page of your site, it asks for one file: `https://yoursite.com/robots.txt`. What it finds there decides where it goes next.

It is the smallest, oldest, and most misunderstood file in web publishing — and one of the few where a single wrong line can quietly cost you all your search traffic.

---

## What robots.txt actually is

A plain text file at the root of your domain that tells automated visitors which parts of the site they should not crawl.

Two things beginners almost always get wrong:

1. **It is a request, not a lock.** Well-behaved crawlers (Google, Bing, Anthropic, OpenAI) obey it. Malicious scrapers ignore it entirely. It is a sign on the lawn, not a fence.
2. **It does not hide anything.** `robots.txt` is public. Anyone can read yours. Listing `/secret-admin/` in it is an advertisement, not a defence.

> **Never put anything in robots.txt that you need to keep private.** You are publishing a list of the paths you care about to the entire internet.

## Where it goes

It must sit at the **root of the host**, and only that location counts:

| URL | Works? |
|---|---|
| `https://example.com/robots.txt` | Yes |
| `https://example.com/blog/robots.txt` | No — ignored |
| `https://blog.example.com/robots.txt` | Yes, but only governs `blog.example.com` |

Subdomains are separate hosts. `example.com/robots.txt` says nothing about `cms.example.com`, which is exactly why a CMS subdomain needs its own file.

## The syntax

There are only a handful of directives.

```
User-agent: *
Disallow: /api/
Allow: /api/public/

User-agent: GPTBot
Allow: /

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

| Directive | Meaning |
|---|---|
| `User-agent` | Which crawler the following rules apply to. `*` means everyone. |
| `Disallow` | A path prefix the crawler should not fetch. Empty value means "nothing is blocked". |
| `Allow` | Carves an exception out of a broader `Disallow`. |
| `Sitemap` | Absolute URL of your sitemap. Not tied to any user-agent. |
| `Crawl-delay` | Seconds between requests. **Google ignores this**; Bing honours it. |

Rules are matched by **longest prefix wins**, not by order. Given a `Disallow: /docs/` and an `Allow: /docs/public/`, the more specific `Allow` wins for that subpath.

Blank lines separate groups. A crawler uses the group that names it specifically, and falls back to `*` only if no group names it — so if you write a `GPTBot` group, it stops reading the `*` group entirely. That trips people up constantly: **rules are not inherited.**

## The mistakes that actually cost traffic

**Blocking CSS and JavaScript.** Google renders your pages like a browser. If you block `/_next/`, `/static/`, or `/assets/`, it sees an unstyled skeleton and may judge the page broken or not mobile-friendly. Google explicitly warns against this.

**Confusing `Disallow` with "don't index".** These are opposites in a way that surprises everyone:

| Goal | Correct tool | Why |
|---|---|---|
| Keep a page out of search results | `noindex` meta tag or `X-Robots-Tag` header | The crawler must **fetch** the page to see the directive |
| Save crawl budget on junk URLs | `Disallow` in robots.txt | Stops the fetch |

If you `Disallow` a page **and** want it deindexed, you have created a contradiction: the crawler can never fetch the page, so it never sees your `noindex`. Google may still list the bare URL because other sites link to it. To remove a page, **allow** the crawl and serve `noindex`.

**A stray leading slash mistake.** `Disallow: /` blocks the entire site. It is one character away from `Disallow:` (blank), which blocks nothing. This single typo is the most common cause of a site vanishing from search overnight.

## AI crawlers

Since 2023 a second population of bots matters. They read robots.txt the same way, under their own user-agent names:

| Crawler | Operator | What it does |
|---|---|---|
| `GPTBot` | OpenAI | Collects data for model training |
| `OAI-SearchBot` | OpenAI | Indexes for ChatGPT Search |
| `ChatGPT-User` | OpenAI | Fetches a page live when a user asks |
| `ClaudeBot` | Anthropic | Crawls for training |
| `Claude-User` | Anthropic | Live fetch on a user's request |
| `PerplexityBot` | Perplexity | Indexes for Perplexity answers |
| `Google-Extended` | Google | **Gemini training only** — does not affect Search |
| `CCBot` | Common Crawl | Public dataset many models are trained on |
| `Bytespider` | ByteDance | Training |

The important nuance: `Google-Extended` **is not Googlebot.** Blocking it keeps you out of Gemini training without touching your Search ranking. Blocking `Googlebot` removes you from Google entirely. Do not confuse them.

Blocking training bots while allowing search bots is a legitimate strategy — you stay findable but opt out of being training data:

```
User-agent: GPTBot
Disallow: /

User-agent: OAI-SearchBot
Allow: /
```

## How to generate one

### Without code

Any of these produce a valid file you paste into your site root:

- Merkle robots.txt generator — clean, no signup
- Ryte robots.txt generator
- SEOptimer generator

Honestly, for most sites the file is five lines. Typing it by hand is faster than any generator.

**Static site / plain HTML** — just create `public/robots.txt` or `/robots.txt`. No build step needed.

## How to test it

1. **Google Search Console → Settings → robots.txt** shows the exact file Google fetched, when, and any parse errors.
2. `curl -s https://example.com/robots.txt` — confirms what is really being served, which is not always what you think you deployed.
3. TametheBots robots.txt tester for checking a specific URL against your rules.

## Checklist

- [ ] Reachable at `https://yourdomain.com/robots.txt`, returns `200` and `text/plain`
- [ ] Does **not** block CSS, JS, or image directories
- [ ] Contains an absolute `Sitemap:` line
- [ ] Every subdomain has its own file (a CMS or staging host needs `Disallow: /`)
- [ ] Nothing private is listed
- [ ] You made a deliberate decision about AI crawlers rather than defaulting

---

## References

- Google: Introduction to robots.txt
- Google: How Google interprets the robots.txt specification
- RFC 9309 — Robots Exclusion Protocol (the 2022 standardisation)
- Google crawlers and user-agent strings
- OpenAI: bots and crawlers
- Anthropic: does Claude crawl the web?