Answer Engine Optimization — Chapter 5: Technical Access

Spencer Goldade

2026

Answer Engine Optimization

Chapter 5: Making Your Site Legible to AI Bots

A free sample chapter from the book

Access, Structure, Authority — The Practitioner's Guide to AI Search Visibility

By Spencer Goldade

This is one chapter of an eighteen-chapter book on getting cited by ChatGPT, Claude, Perplexity, and Google's AI Overviews. It covers the Access layer of the book's three-part framework: whether AI crawlers can reach your content at all. Read this chapter first, it's where most sites lose visibility they didn't know they had.

If the chapter is useful, the full book is at **spencergoldade.ca/aeo**.

Making Your Site Legible to AI Bots

This is the first layer of the Access, Structure, Authority framework, and it's the chapter where you stop reading about AEO and start doing it. **Access** is the most common AEO failure, and it's also the most fixable. If AI bots can't reach your content, Structure and Authority are irrelevant. The good news: most technical access problems can be identified in under an hour and fixed in under a day. This chapter gives you every user agent to know, every blocker to check, and three robots.txt templates matched to three legitimate stances you can take on AI access.

Decide before you configure

Before you touch a robots.txt file or a CDN rule, decide what you actually want. Not every site should maximize AI visibility. A service business whose competitive edge is being findable has a different calculus than a media publisher whose content funds journalism, or a creative studio licensing its work, or a membership community whose members object to having their contributions trained on. Three legitimate stances to pick between:

- **Permit all.** Your content is findable to every AI crawler, including training crawlers. You trade training rights for maximum visibility.
- **Citation-only.** Training crawlers blocked, citation and search crawlers allowed. You keep AI visibility and preserve licensing rights for your content.
- **Block all AI crawlers.** You opt out of AI ingestion entirely. You lose AI citation coverage; that's the trade.

The templates at the end of this chapter implement all three. Most of what follows applies regardless of which stance you pick. The access problems are the same, the defensive tooling is the same, only the rules you write change. Decide early; you'll read the rest of the chapter differently once you know which template you're heading toward. Chapter 11 names these costs in full.

The silent blocker: how Cloudflare is hiding you from AI search

I'm leading with this because it's the problem most sites don't know they have.

Before I wrote this chapter, I ran an audit of 50 B2B SaaS websites. I expected most of them to be blocking AI bots in their robots.txt. That was my strongest prior walking in. It was also wrong. Only 2 of the 50 sites blocked AI bots at the robots.txt layer — 96% explicitly allowed them. I was wrong about where the blocking happens. But I was right that most sites are blocking AI bots somewhere, because during that same audit, 5 sites couldn't be crawled at all. One of them was Linear, whose Cloudflare challenge wall blocked my research crawler before it could read a single page (**April 2026** crawl sample). That's the same class of wall that can block AI crawlers when sites treat them like generic bots.

Cloudflare's Bot Fight Mode is a baseline bot-protection feature on the Free plan, and it can be toggled on paid plans as well. Many Cloudflare-fronted sites have it active, often without anyone on the team remembering it was turned on. It was designed to block malicious bots: scrapers, credential stuffers, DDoS traffic. But it does not cleanly distinguish between bad bots and AI search crawlers. When Bot Fight Mode is active, it can silently block GPTBot, ClaudeBot, PerplexityBot, and other AI crawlers. Your server never sees the request. Your logs show nothing. You have no idea it is happening.

This matters because many sites route through Cloudflare (W3Techs and similar surveys routinely show it among the largest reverse-proxy/CDN footprints, **EMERGING** exact percentage). If your site is behind Cloudflare (check your response headers for `cf-ray` or `server: cloudflare`), Bot Fight Mode or related bot rules are a common place to find AI crawlers blocked, even when `robots.txt` looks permissive. Your robots.txt can say "all bots welcome" and the CDN layer can still be quietly turning them away.

How to check:

1. Log into your Cloudflare dashboard
2. Navigate to Security → Bots
3. Check whether Bot Fight Mode is enabled
4. If it is, you have two options:

Option A (recommended): Disable Bot Fight Mode entirely if you're on the free plan. On paid plans, use Super Bot Fight Mode with custom rules that allow known AI crawlers.

Option B: Create WAF (Web Application Firewall) custom rules that explicitly allow AI crawler user agents. This is more granular but requires maintaining the rule list as new crawlers appear.

Priority: Critical. This is a 15-minute fix that can immediately unblock all AI crawler traffic to your site.

The fixes in this chapter assume some level of technical control: CDN dashboard access, WAF rules, server configuration, or a developer on retainer. If you run your site on a hosted

platform with no custom-firewall access (Wix Personal, Squarespace Personal, basic GoDaddy), or if you're a solopreneur without a developer to call, you can't implement several of these fixes at all. That's a real gap. Focus on the levers you do have (robots.txt, clean HTML, view-source testing) and don't waste time on CDN-level changes you can't make. If you're an agency advising a small-budget client, be honest about which recommendations apply and which are enterprise-only.

The major AI user agents (and what each one does)

AI crawlers identify themselves through user-agent strings in their HTTP requests. The list below covers the platforms most relevant to English-language AEO as of early 2026. It is not exhaustive. New crawlers appear regularly, and the list skews toward North American and European platforms because that's where the documentation is most accessible.

OpenAI (ChatGPT)

- **GPTBot:** OpenAI's primary crawler. Used for training data collection and general indexing. If you allow GPTBot, your content may be used to train future models AND be available for citation.
- **OAI-SearchBot:** OpenAI's search-specific crawler. Used for real-time citation retrieval when ChatGPT searches the web. Allowing this is how your content gets cited in ChatGPT responses.
- **ChatGPT-User:** The user-agent for real-time fetches triggered by a ChatGPT user's query. This is the bot that visits your site when someone asks ChatGPT a question and it needs to check your page.

Key distinction: You can block GPTBot (training) while allowing OAI-SearchBot and ChatGPT-User (citation). This lets you opt out of training data contribution while remaining citable. Before you decide, consider whose work is on your site. If your blog contains articles from contract writers, guest contributors, or former employees who weren't paid with AI training in mind, allowing GPTBot means their work trains commercial models without renewed consent or compensation. Many publishers and content companies use the split-allow pattern (block training, allow citation) to stay visible in AI search while preserving those writers' position for any future licensing arrangement.

Note that OpenAI's user agent landscape is still evolving. New user agents may appear as OpenAI launches new products. Check OpenAI's documentation quarterly to stay current.

Anthropic (Claude)

- **ClaudeBot:** Anthropic's primary crawler for indexing and citation.

- **anthropic-ai:** Anthropic's training data crawler.
- **claude-web:** Anthropic's web-focused crawl agent.

Perplexity

- **PerplexityBot:** Perplexity's indexing crawler. It builds Perplexity's independent web index.
- **Perplexity-User:** Triggered by human-initiated Perplexity searches. Similar to ChatGPT-User.

Google

- **Google-Extended:** Google's AI/Gemini data crawler. Separate from Googlebot (which handles traditional search). Blocking Google-Extended blocks AI training use while preserving your Google search rankings.

Microsoft

- **Bingbot:** Bing's main crawler, also used by Microsoft Copilot. If you already allow Bingbot for Bing search, Copilot gets your content too.

Others

- **Applebot-Extended:** Apple's AI crawler for Apple Intelligence and Siri.
- **Amazonbot:** Amazon's crawler, used for Alexa and Amazon services.
- **FacebookBot:** Meta's crawler for content indexing.
- **meta-externalagent:** Meta's AI training crawler.

International AI platforms

If your audience extends beyond English-speaking markets, be aware that major AI platforms in other regions have their own crawlers. Baidu's ERNIE (China), DeepSeek (China), Naver's HyperCLOVA (South Korea), and Yandex's AI features (Russia) all operate retrieval systems, but their crawler user agent documentation is less centralized than what OpenAI or Google publish. If you serve international markets, check your server logs for unfamiliar bot traffic and research the user agents you find. The AEO principles in this book apply regardless of platform, but the specific user agent strings will differ.

Why AI bots can't read your JavaScript (and what to do about it)

Most AI crawlers **do not execute JavaScript the way Chrome does**. They read the raw HTML your server returns on the initial request. As of early 2026, the major retrieval bots (GPTBot, ClaudeBot, PerplexityBot, claude-web) do not publish browser-grade rendering. Treat client-rendered pages as at risk until you verify with `curl` and vendor docs (**EMERGING** as vendors evolve).

This is a critical difference from Googlebot, which can render JavaScript (though it does so on a delay). If you have built your site assuming JavaScript rendering, your content may be visible to Google but invisible to the AI crawlers in practice.

An important framing note: this isn't only a problem to fix. It can also be a tool. If you have content you want humans to see but prefer AI not to index (proprietary tools, gated community content, interactive experiences where the context matters), client-side rendering is one way to achieve that selectively. The same mechanics that make a React SPA invisible to AI crawlers can be a deliberate choice, not an accident. The rest of this section assumes you want to maximize AI visibility, but keep in mind that the same knowledge works in reverse when you need it to.

A "SPA" (Single Page Application) is a website built as a single HTML shell where all the content loads dynamically through JavaScript after the page arrives in your browser. Frameworks like React, Vue, and Angular commonly build SPAs. They feel fast to human visitors, but to an AI crawler that only reads the initial HTML, the page looks empty.

Content that AI bots can't see:

- React SPAs without server-side rendering
- Vue.js or Angular apps that render content client-side
- Content inside tabs, accordions, sliders, or modals that require clicks to reveal
- Dynamically loaded content (infinite scroll, lazy-loaded sections)
- Pricing calculators, interactive tools, or configurators
- Content behind login walls or paywalls

How to test what AI bots see:

1. Open your browser and navigate to your key pages
2. Right-click and select "View Page Source" (not "Inspect Element")
3. Search the raw HTML for your important content
4. If your content isn't in the raw HTML, AI bots can't see it

How to fix it:

If you're on **Next.js**: Enable SSR (Server-Side Rendering) or SSG (Static Site Generation) for content pages. Check your `next.config.js` to confirm. Pages using `getServerSideProps` or `getStaticProps` are server-rendered. Pages that rely only on client-side data fetching (useEffect with fetch) are not.

If you're on **Nuxt.js**: Similar to Next.js. Use `asyncData` or `useFetch` in setup for server-side data loading. Confirm SSR is enabled in `nuxt.config.js`.

If you're on **Create React App or Vite (no SSR)**: This is the hardest case. Your options are migrating to Next.js, adding a prerendering layer (like Prerender.io or react-snap), or converting critical content pages to static HTML.

If you have **content behind tabs or accordions**: Move the content into the visible page body. If you must use tabs for UX reasons, make sure all tab content is present in the initial HTML and only hidden via CSS (display: none), not loaded on click via JavaScript. AI crawlers won't click, but they will read hidden HTML.

Priority: Critical for SPAs. Important for sites with interactive content elements.

A note on headless CMS platforms (Contentful, Sanity, Strapi): these architectures are compatible with AEO provided your frontend framework renders content server-side. The CMS itself has no bearing on AI crawler access because crawlers interact with the rendered output, not the content management layer. The rendering strategy of your frontend determines visibility. If your frontend generates static HTML at build time (the Jamstack pattern), your content is fully accessible to crawlers. If it operates as a client-side SPA that fetches content through API calls after the initial page load, that content remains invisible to AI retrieval systems regardless of which CMS stores it upstream.

The diagnostic test is consistent across all frameworks: View Page Source in your browser. If the substantive content appears in the raw HTML response, your rendering configuration is adequate. If the HTML contains only a shell application with JavaScript bootstrap code, you have a rendering problem that must be resolved before other AEO work can take effect.

Rate limiting and crawl throttling

Some sites block AI bots by accident through overly aggressive rate limiting, and the symptoms are subtle enough that most teams never notice.

If your server or CDN enforces per-IP request limits (a standard anti-abuse measure), AI crawlers can trip those thresholds during routine indexing passes. The crawler receives a 429 response after a handful of pages, abandons the crawl, and moves on to the next domain. Your most valuable content never gets indexed, and your server logs may not retain enough detail to surface the pattern.

How to check:

1. Look for rate-limiting configuration in your server settings (nginx `limit_req`, Apache `mod_ratelimit`, Cloudflare Rate Limiting rules)
2. Check if the limits differentiate between human visitors and known bot user agents
3. Review server logs for 429 (Too Many Requests) responses to AI crawlers

How to fix:

If you're using Cloudflare: create a WAF rule that exempts known AI crawler user agents from rate limits. If you're managing rate limiting at the server level: add exceptions for the AI user agent strings listed earlier in this chapter.

This is a lower-priority issue than robots.txt or Bot Fight Mode, but it can silently reduce your AI crawl coverage over time.

Content behind paywalls and login walls

Content behind authentication is invisible to AI crawlers. This is by design rather than a bug, but the downstream consequences catch many content-driven businesses off guard.

If your highest-value content (the original research, detailed guides, and competitive analysis that would generate the strongest AI citations) lives behind a login or paywall, AI retrieval systems cannot access it during their indexing passes. They cannot cite what they cannot read, and they cannot discover gated content through secondary references alone.

The trade-off is real and involves competing business objectives. Gating content generates leads and qualifies purchase intent. Un-gating content generates AI citations and builds brand visibility in a channel where you cannot buy placement. There is no universal right answer, but consider a hybrid approach that serves both goals:

1. **Un-gate your definitional content.** Pages that answer "what is X" and "how does Y work" should be freely accessible. These are the pages most likely to match AI sub-queries.
2. **Gate your proprietary research and tools.** Original data, interactive calculators, and premium reports can remain gated. The free definitional content establishes your authority; the gated content captures leads.
3. **Offer excerpt access.** Some sites show the first 500-1,000 words of gated content before requiring login. AI crawlers can extract the opening (which should follow the BLUF principle) and cite the page even if they can't read the full content.

HTTPS and security headers

All AI crawlers respect HTTPS, and several retrieval systems will either skip HTTP-only sites entirely or assign them reduced trust during the citation-selection process. If your site is still serving pages over unencrypted HTTP in 2026, resolve that prerequisite before addressing anything else in this chapter.

Security headers introduce another layer of potential interference with AI crawler access. Review your HTTP response headers for the following:

- **X-Robots-Tag:** This HTTP header can block specific crawlers even if your robots.txt allows them. Check for `X-Robots-Tag: noindex` headers that might apply to AI user agents.
- **Content-Security-Policy:** Overly strict CSP headers can interfere with how some crawlers parse your page, though this is rare.

The staging-freeze problem that's blocking thousands of sites

This one is embarrassingly common.

A developer adds `Disallow: /` to robots.txt during a staging or pre-launch freeze to prevent search engines from indexing unfinished content. The site launches successfully. The development team moves on to post-launch priorities. Nobody revisits the robots.txt configuration. Months or years later, the site is still broadcasting a blanket crawl prohibition to every bot that checks, including every AI retrieval crawler.

I've encountered this pattern on agency sites, enterprise SaaS platforms, and small business sites alike. The problem is structurally invisible: no error appears in your analytics, no alert fires in your monitoring, and the pages continue to render normally for human visitors.

How to check:

1. Visit `yoursite.com/robots.txt` in a browser
2. Look for `User-agent: *` followed by `Disallow: /`
3. If you see it, every bot (Google, AI, all of them) is being told not to crawl any page on your site

The fix takes 30 seconds. But finding the problem requires knowing to look.

A related issue: robots.txt files that block specific directories where your best content lives. Some CMS platforms put blog posts under `/blog/` or resources under `/resources/`. If your robots.txt has `Disallow: /blog/` from a legacy configuration, your entire content library is

invisible to AI crawlers. Check every Disallow line against the directories you actually want indexed.

Another common variant: a `noindex` meta tag left on key pages. This is separate from robots.txt. Even if your robots.txt allows crawling, a `<meta name="robots" content="noindex">` tag on the page itself tells crawlers not to index it. View source on your important pages and search for "noindex" to be sure.

How to audit your AI crawler access in 30 minutes

Here's the complete audit procedure, step by step.

Step 1: Check robots.txt (5 minutes)

1. Navigate to `yoursite.com/robots.txt`
2. Look for any blanket `Disallow: /` rules
3. Check each AI user agent listed above: is it explicitly allowed, blocked, or not mentioned?
4. If a user agent isn't mentioned, the default `User-agent: *` rules apply

Step 2: Check Cloudflare or CDN settings (5 minutes)

1. Check your response headers for Cloudflare indicators (`cf-ray`, `server: cloudflare`)
2. If Cloudflare: log into the dashboard and check Bot Fight Mode
3. If another CDN or WAF: check bot-management settings for AI crawler blocking

Step 3: Check JavaScript rendering (10 minutes)

1. Open your top 5 pages in a browser
2. View Page Source on each
3. Search the source for your key content (headlines, product descriptions, FAQ answers)
4. Flag any page where critical content is missing from the source HTML

Step 4: Check server logs for AI bot traffic (10 minutes)

If you have access to server logs, search for AI crawler visits:

```
grep -i "gptbot\|claudebot\|perplexitybot\|oai-searchbot" access.log
```

If you see zero AI bot traffic after removing blocks, give it 1-2 weeks for crawlers to return and check again. If you still see nothing after two weeks, check for CDN-level blocking or rate limiting.

Step 5: Check page speed (5 minutes)

Run your top 3 pages through Google PageSpeed Insights. Flag any page with LCP above 2.5 seconds. AI crawlers appear to be less patient with slow servers than Googlebot, based on practitioner reports. Slow pages get skipped more readily.

XML sitemaps and AI crawlers

Your XML sitemap serves the same discovery function for AI crawlers that it serves for Googlebot, providing a structured inventory of indexable URLs with modification timestamps. But several implementation details carry additional weight in an AEO context.

1. **Keep your sitemap current.** AI crawlers use sitemaps to find new and updated content. A stale sitemap that hasn't been regenerated in months means AI crawlers may miss your recent pages.
2. **Include lastmod dates.** AI systems weight freshness. If your sitemap includes accurate `<lastmod>` timestamps, crawlers can prioritize recently updated content.
3. **Don't include pages you've blocked in robots.txt.** This sends contradictory signals.
4. **Keep it under 50,000 URLs per file.** This is the standard limit. Use sitemap index files for larger sites.

Submit your sitemap to Google Search Console. AI crawlers do not consume Search Console data directly, but Google's index feeds into ChatGPT's retrieval pipeline (via the SERP API integration discussed in Chapter 3), so a properly submitted and validated sitemap indirectly improves your AI citation coverage.

Common mistakes

- **Assuming your robots.txt is fine because Google indexes you.** Google uses Googlebot. AI platforms use different user agents. You can be indexed by Google and blocked from every AI crawler.
- **Not checking Cloudflare.** Bot Fight Mode is the most common invisible blocker, and most site owners don't know it's on.
- **Testing with browser DevTools instead of View Source.** DevTools shows the rendered page (after JavaScript runs). View Source shows what AI bots actually see.

- **Blocking training crawlers and citation crawlers together.** You can block GPTBot (training) while allowing OAI-SearchBot (citation). They're separate user agents.
- **Forgetting to re-check after changes.** Crawlers take 1-4 weeks to return after you remove blocks. Don't assume instant results.

Chapter Summary

- Cloudflare Bot Fight Mode is enabled by default and silently blocks AI crawlers. Check your Cloudflare dashboard immediately.
- AI crawlers identify via user-agent strings. The major ones: GPTBot, OAI-SearchBot, ChatGPT-User, ClaudeBot, PerplexityBot, Google-Extended, Bingbot.
- You can separate training access (GPTBot, anthropic-ai) from citation access (OAI-SearchBot, ChatGPT-User) in your robots.txt.
- AI bots don't execute JavaScript. Content rendered client-side (React SPAs, Vue apps, interactive elements) is invisible to AI crawlers.
- The staging-freeze robots.txt problem (blanket Disallow: /) is common and easy to miss. Check yours now.
- A complete AI crawler access audit takes about 30 minutes: robots.txt, Cloudflare, JS rendering, server logs, and page speed.

Chapter Checklist

☐ Check `yoursite.com/robots.txt` for blanket `Disallow: /` rules blocking all bots

☐ Verify each AI user agent (GPTBot, OAI-SearchBot, ClaudeBot, PerplexityBot, Google-Extended) is not blocked

☐ Check Cloudflare dashboard for Bot Fight Mode (Security → Bots) and disable or configure exceptions

☐ View Page Source on your top 5 pages and confirm critical content appears in the raw HTML

☐ If running a React/Vue/Angular SPA, verify SSR or SSG is enabled for content pages

☐ Move content out of tabs, accordions, and modals into the main page body (or ensure it's in the initial HTML)

☐ Search server logs for AI bot user agents to confirm they're visiting

☐ Run PageSpeed Insights on top 3 pages and fix any LCP scores above 2.5 seconds

☐ If using a WAF other than Cloudflare, check bot-management rules for AI crawler blocking

Ready-to-Use Artifacts: robots.txt templates for AI Crawlers

There is no universally correct robots.txt for AI crawlers. The template you pick is a statement about what rights you want to preserve for the content on your site. Below are three options, each legitimate for different contexts. All three preserve your traditional search behaviour. They only change how AI-specific crawlers are treated, so your Google and Bing rankings don't move when you switch.

Permit all. Allow every crawler, including training. Right for service businesses, public-facing brand sites, and anyone whose competitive edge is being findable. You trade training rights for maximum visibility.

Citation-only (publisher-standard). Block training crawlers, allow citation and search crawlers. Right for media companies, research publishers, creative work, and anyone with licensing interests to preserve. This is what most major publishers have adopted in 2024–2026. You keep AI visibility and retain the ability to license your content separately.

Block all AI crawlers. Opt out entirely. Right for sites whose audiences explicitly object to AI ingestion, or sites where the work's commercial life depends on controlled distribution. You lose AI citation coverage; that's the trade.

Adjust Allow/Disallow paths in any of the below to match your site structure.

Template 1: Permit all

```
# ==========================================
# AI CRAWLER ACCESS CONFIGURATION — PERMIT ALL
# Last updated: April 2026
# ==========================================

# --- OpenAI ---
# GPTBot: Training data crawler. Allow if you want your content
# used for training AND available for citation.
# Block if you only want citation access (see Template 2).
User-agent: GPTBot
Allow: /

# OAI-SearchBot: Citation crawler. This is how your content
# appears in ChatGPT responses. Allow this unless you want to
# be invisible to ChatGPT entirely.
User-agent: OAI-SearchBot
Allow: /

# ChatGPT-User: Real-time fetch when a user asks ChatGPT
# a question. Blocking this prevents live citation.
User-agent: ChatGPT-User
Allow: /

# --- Anthropic (Claude) ---
User-agent: ClaudeBot
Allow: /

User-agent: anthropic-ai
Allow: /

User-agent: claude-web
Allow: /

# --- Perplexity ---
User-agent: PerplexityBot
Allow: /

User-agent: Perplexity-User
Allow: /

# --- Google AI / Gemini ---
# Google-Extended controls AI/Gemini use of your content.
# Separate from Googlebot (traditional search).
# Blocking Google-Extended does NOT affect your Google rankings.
```

```
User-agent: Google-Extended
Allow: /

# --- Microsoft Copilot ---
# Copilot uses Bingbot. If you allow Bingbot for Bing search,
# Copilot gets your content too.
User-agent: Bingbot
Allow: /

# --- Apple ---
User-agent: Applebot-Extended
Allow: /

# --- Amazon ---
User-agent: Amazonbot
Allow: /

# --- Meta ---
User-agent: FacebookBot
Allow: /

User-agent: meta-externalagent
Allow: /

# --- Default ---
# Keep your existing rules for all other bots.
User-agent: *
Allow: /
Disallow: /admin/
Disallow: /api/
Disallow: /staging/

# Sitemap
Sitemap: https://yoursite.com/sitemap.xml
```

This template is also available in the AEO Supporting Files bundle from spencergoldade.ca/aeo as `robots-permit-all.txt`. The full annotated version is also in Appendix A.

Template 2: Citation-only (publisher-standard)

Blocks training-specific crawlers while letting citation and search crawlers through. Most major publishers have adopted this pattern in 2024–2026 to keep licensing rights intact without becoming invisible to AI search.

```
# ==========================================
# AI CRAWLER ACCESS CONFIGURATION — CITATION ONLY
# Last updated: April 2026
# ==========================================

# --- Block training crawlers ---
User-agent: GPTBot
Disallow: /

User-agent: anthropic-ai
Disallow: /

User-agent: Google-Extended
Disallow: /

User-agent: Applebot-Extended
Disallow: /

User-agent: meta-externalagent
Disallow: /

User-agent: CCBot
Disallow: /

# --- Allow citation and search crawlers ---
User-agent: OAI-SearchBot
Allow: /

User-agent: ChatGPT-User
Allow: /

User-agent: ClaudeBot
Allow: /

User-agent: PerplexityBot
Allow: /

User-agent: Perplexity-User
Allow: /

User-agent: Bingbot
Allow: /

# --- Default ---
```

```
User-agent: *
Allow: /
Disallow: /admin/
Disallow: /api/
Disallow: /staging/

# Sitemap
Sitemap: https://yoursite.com/sitemap.xml
```

This template is also available in the AEO Supporting Files bundle from spencergoldade.ca/aeo as `robots-citation-only.txt`. The full annotated version is in Appendix A.

Template 3: Block all AI crawlers

Blocks every known AI crawler while keeping traditional search crawlers (Googlebot, regular Bingbot for search) free to index normally. Your Google and Bing search rankings are unaffected.

```
# ==========================================
# AI CRAWLER ACCESS CONFIGURATION — BLOCK ALL
# Last updated: April 2026
# ==========================================

# --- OpenAI ---
User-agent: GPTBot
Disallow: /

User-agent: OAI-SearchBot
Disallow: /

User-agent: ChatGPT-User
Disallow: /

# --- Anthropic ---
User-agent: ClaudeBot
Disallow: /

User-agent: anthropic-ai
Disallow: /

User-agent: claude-web
Disallow: /

# --- Perplexity ---
User-agent: PerplexityBot
Disallow: /

User-agent: Perplexity-User
Disallow: /

# --- Google AI / Gemini (training only) ---
User-agent: Google-Extended
Disallow: /

# --- Apple ---
User-agent: Applebot-Extended
Disallow: /

# --- Amazon ---
User-agent: Amazonbot
Disallow: /
```

```
# --- Meta ---
User-agent: FacebookBot
Disallow: /

User-agent: meta-externalagent
Disallow: /

# --- ByteDance ---
User-agent: Bytespider
Disallow: /

# --- Common Crawl (feeds many AI systems) ---
User-agent: CCBot
Disallow: /

# --- Default: keep normal web behaviour ---
User-agent: *
Allow: /
Disallow: /admin/
Disallow: /api/
Disallow: /staging/

# Sitemap
Sitemap: https://yoursite.com/sitemap.xml
```

This template is also available in the AEO Supporting Files bundle from spencergoldade.ca/aeo as `robots-full-block.txt`. The full annotated version is in Appendix A.

After you deploy

After saving any of these templates to `/robots.txt`, verify it's working. Two fast checks:

- Fetch your robots.txt directly: `curl https://yoursite.com/robots.txt` and confirm the rules appear as you wrote them.
- Use Google's robots.txt Tester (in Google Search Console) to confirm your syntax parses correctly.

For ongoing verification, Chapter 15 covers how to analyse server logs for AI crawler traffic (privacy-first patterns). After you switch templates, you should see the allowed crawlers in your logs within a few weeks as they re-crawl; blocked crawlers drop off on their next visit.

What's in the rest of the book?

This chapter covers Access, the first of three layers. The other two layers, and the rest of the book, go here.

What is the Access, Structure, Authority framework?

Three tests your content has to pass to get cited by AI. **Access** is whether AI crawlers can reach your content (this chapter). **Structure** is whether they can extract a clean answer once they arrive. **Authority** is whether they'll cite you over someone else. Each layer fails silently, and each one has to be fixed in order. Working on Structure before Access is effort spent on pages AI bots never see.

Who is the full book for?

SEO practitioners, digital marketers, content leads, and agency owners who need to make AEO decisions this quarter. It's written practitioner-to-practitioner, not as a field survey. Every chapter gives you something to change on your site this week.

What else does the full book cover?

- llms.txt, schema markup, and content structure (Access + Structure layers in depth)
- E-E-A-T, freshness, information gain, and entity optimization (Authority layer)
- Third-party platforms: Reddit, YouTube, LinkedIn, and the citation-density problem
- Platform playbooks: what actually works on ChatGPT, Claude, Perplexity, and Google AI Overviews
- Measurement: server logs, GA4 channel setup, and first-party AI visibility reports
- A 90-day action plan with a week-by-week tracker
- Six appendices: robots.txt templates, llms.txt templates, schema library, audit checklist, glossary, tools reference

Where do I get it?

Everything ships from **spencergoldade.ca/aeo**. Three options:

- **Ebook only** — the full manuscript in EPUB and PDF

- **Supporting files only** — robots.txt templates, llms.txt templates, schema JSON library, audit checklists, tracker spreadsheets, and the 90-day action plan (four formats)
- **Bundle** — ebook and supporting files together at a reduced price

Amazon and other retail readers get the ebook at their chosen retailer. The supporting files bundle is available separately at spencergoldade.ca/aeo (with two free samples — the hub-and-spoke worksheet and the 90-day action plan — at spencergoldade.ca/aeo-samples).

What should I do before I finish this PDF?

- ☐ Check your site for a `cf-ray` response header. If present, audit Cloudflare Bot Fight Mode.
- ☐ Fetch your site with `curl -A "GPTBot"` and confirm you get a real response.
- ☐ Pick one of the three robots.txt stances in this chapter (permit all, citation-only, block all) and write it down before you configure anything.
- ☐ When you're ready for Structure and Authority, pick up the book (or book + supporting files) at spencergoldade.ca/aeo.

Spencer Goldade is a product and design leader based in Calgary, Alberta. He came to Answer Engine Optimization from product design and entrepreneurship, and built a custom audit tool to run against 50 B2B SaaS sites for the original research cited throughout the full book.