Geo Ready

FREE PREVIEW · 3 CHAPTERS

# The GEO Readiness Manual

A Technical Guide to AI Search Visibility — Free Preview (3 of 15 chapters)

# Chapter 0: The Selection Layer - Why AI Search Became a Filter

## AI search is not another traffic channel

AI search is not a new version of organic search with a different interface. It is a selection layer between the user and the web. A traditional search engine exposes many possible sources and asks the user to choose. An AI answer engine chooses first, then shows the user a synthesized answer built from a small set of retrieved sources.

That change is brutal for publishers, SaaS companies, consultants, e-commerce brands, and technical documentation teams. In classic SEO, losing position 1 usually meant losing traffic gradually. In AI search, not being selected can mean disappearing from the answer entirely. The user may never see the omitted sources, because the interface has already compressed the choice set before the user starts comparing.

This is the core tension behind Generative Engine Optimization. GEO is not about tricking a model into mentioning a brand. GEO is about making a site eligible, legible, and defensible enough that a retrieval system can select it when the answer needs evidence.

## The web page is no longer the unit of competition

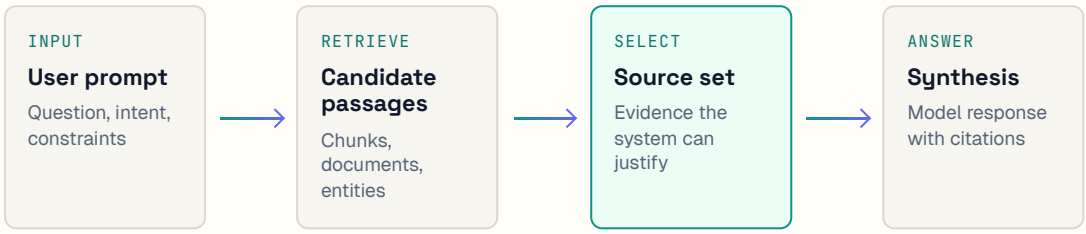
Classic SEO made the page the main object of work: one URL, one keyword set, one ranking position, one click. AI answer engines operate differently. They retrieve passages, resolve entities, compare source confidence, and synthesize an answer from fragments. A page still matters, but the model often scores smaller units than the page and larger units than the page at the same time.

At the smaller level, a single paragraph can win retrieval if it answers the prompt more directly than a competitor's page. At the larger level, a brand can lose citation even with a good paragraph if the engine cannot resolve the brand entity, cannot connect the author to the publication, or sees stronger independent confirmation for a competitor.

That is why GEO has to work across layers. A page with a quotable paragraph but no crawl access is invisible. A crawlable page with invalid schema is ambiguous. A structured page with no external entity presence looks self-declared. A known entity with vague content gives the model nothing precise to cite. The work is not one tactic. It is a chain.

**Diagram 1 - The AI selection layer**

AI search compresses discovery into a retrieval and citation pipeline before the user sees the answer.




## The model is not your reader

A human reader can forgive friction. A human can scroll past a cookie banner, wait for JavaScript, interpret a clever headline, infer that "it" refers to the product named three paragraphs earlier, and understand a chart even if the key number is not written in text. A crawler and retrieval pipeline may not do any of that.

AI systems operate through lossy transformations. HTML becomes text. Text becomes chunks. Chunks become embeddings. Embeddings become candidates. Candidates become a context window. The model then writes an answer from whatever survived the pipeline. Every unclear heading, missing entity name, invalid JSON-LD block, JavaScript-only paragraph, duplicate URL, and weak external mention increases the chance that the wrong thing survives.

This is why GEO writing often feels stricter than normal editorial writing. The goal is not to make content robotic. The goal is to remove ambiguity before the machine has to guess. Good GEO content is still written for humans, but it is structured so a machine can extract the same meaning a human would.

## There is no clean boundary between technical SEO and content strategy

GEO punishes organizational silos. The developer controls rendering, status codes, robots.txt, CDN rules, and schema injection. The writer controls headings, passage clarity, examples, definitions, and factual density. The SEO or growth lead controls query sets, competitor benchmarks, internal links, and off-domain entity signals. The founder or product lead controls positioning, pricing truth, roadmap claims, and what the company is willing to say publicly.

If any one of those roles treats GEO as someone else's job, the system leaks. A content team can write the strongest guide in the category and still fail if the site ships a JavaScript shell. A developer can ship perfect static HTML and schema, but the model will still skip the page if

the content says nothing concrete. A growth team can track citations every week, but measurement without fixes becomes reporting theater.

The practical discipline is simple: fix access first, then orientation, then understanding, then quotability, while building entity authority over time. The order matters because every layer depends on the one before it.

## What GeoReady measures, and what it cannot promise

GeoReady measures readiness, not guaranteed placement. The audit engine checks whether a public URL exposes the signals that AI answer engines can plausibly use: crawler access, server-rendered content, metadata, schema, llms.txt, headings, factual density, AI discovery files, brand entity consistency, and security patterns that can pollute extraction.

Those signals do not create a contract with ChatGPT, Perplexity, Claude, Google, Gemini, Apple, or any other answer engine. Retrieval systems are black boxes, model updates change behavior, and the same prompt can produce different answers across sessions, regions, and engines. A high GEO score increases eligibility and reduces obvious blockers. It does not guarantee citation.

This distinction matters commercially. If a vendor promises guaranteed AI citations, the vendor is selling certainty that the market does not provide. A serious GEO practice should be auditable, repeatable, and honest about uncertainty. The work is to improve the odds, measure the result, and keep improving the weak layer.

## Why this guide is intentionally severe

This guide does not treat GEO as a branding trend. It treats GEO as infrastructure for being selected by machines that mediate demand. That makes the standard higher than a blog checklist. Every recommendation has to survive three questions: can the crawler access it, can the retrieval system parse it, and can the model justify citing it?

If a tactic cannot pass those questions, it does not belong in a serious GEO workflow. llms.txt is useful only if it points to crawlable, valuable pages. Schema is useful only if it matches visible content. Content is useful only if it says something specific. Benchmark data is useful only if the methodology is clear enough to challenge.

That is the posture of this manual: technical, critical, and practical. GEO is young. The engines are opaque. But the web still gives us enough observable surface to work carefully: server logs, source HTML, structured data, public citations, prompt sets, competitor mentions, and longitudinal drift. Use those surfaces. Ignore shortcuts.

---

*This is an excerpt from The GEO Readiness Manual. The full 160-page manual continues with Chapter 4 (Access) and Chapter 7 (Quotability) in this preview.*

---

# Chapter 4: Access - The Zero Layer

## Why access comes before everything

In the four-layer model - access, orientation, understanding, quotability - access is the zero layer. Not because it's the most sophisticated, but because if it fails, every other layer is irrelevant. You can have perfect schema, brilliant content, and a strong entity presence. If the AI crawler receives an empty page, a 403, or a JavaScript shell with no content, none of it matters.

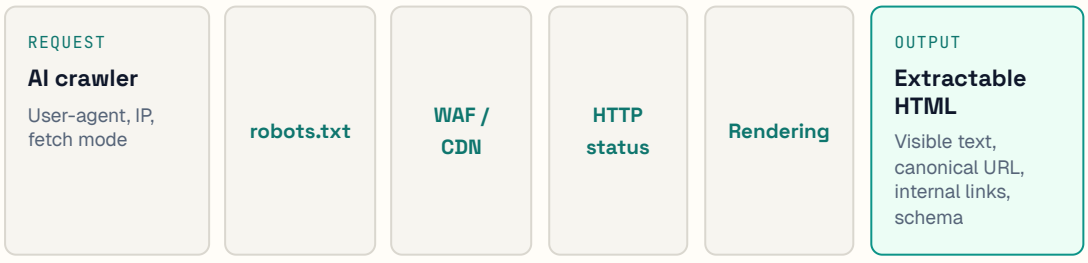
We see this in audit data constantly. A site owner invests in content, hires writers, publishes weekly. The content is excellent. The schema is valid. The internal linking is clean. But the site renders content only through client-side JavaScript, and the AI crawler's basic fetch sees a blank page. The site is invisible to every AI answer engine, and the owner can't understand why.

Access is not glamorous. It's infrastructure. But it's the layer where the most damaging mistakes happen, and where the fixes are often the simplest.

This chapter covers everything between "AI crawler attempts to fetch your page" and "AI crawler receives your content in a parseable form." That includes robots.txt (already covered in chapter 3, but expanded here with edge cases), rendering, server responses, HTTP headers, middleware, and the obstacles that sit between the crawler and your content.

**Diagram 3 - AI crawler access map**

Access failures usually happen before content quality can matter.




If any gate returns a block, empty shell, redirect loop, or ambiguous canonical, downstream GEO work becomes invisible.

## robots.txt: beyond the basics

Chapter 3 covered the crawler reference and configuration patterns. Here we address the edge cases that cause silent failures.

### Directive order and specificity

robots.txt evaluates directives top-to-bottom, but uses specificity matching for user-agents. If you have:

```
User-agent: *
Disallow: /private/

User-agent: GPTBot
Allow: /
```

GPTBot matches its specific rule (Allow: /) and ignores the wildcard. But if the order were reversed and GPTBot's rule came first, the result would be the same - specificity, not order, determines which rule applies for a given user-agent.

The trap: if you have multiple specific rules and a wildcard, ensure the wildcard doesn't accidentally override a specific allow with a more specific disallow path:

```
User-agent: *
Allow: /

User-agent: GPTBot
Disallow: /api/
```

GPTBot is allowed everything except /api/. Correct. But:

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

User-agent: GPTBot
Allow: /
```

GPTBot is allowed everything - including /api/ - because the specific `Allow: /` overrides the wildcard `Disallow: /api/`. If you intended to block /api/ for all crawlers including GPTBot, you need:

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

User-agent: GPTBot
Disallow: /api/
Allow: /
```

### Wildcard patterns

robots.txt supports two wildcard characters: `*` (any sequence of characters) and `$` (end of URL).

```
User-agent: GPTBot
Disallow: /*.pdf$
Disallow: /*?session=
Disallow: /search?
```

- `/*.pdf$` blocks all PDF files.
- `/*?session=` blocks all URLs containing `?session=` anywhere.
- `/search?` blocks all search result pages (useful to prevent crawling of internal search).

Use wildcards to prevent AI crawlers from indexing low-value pages: search results, faceted navigation, session-based URLs, duplicate parameter combinations. These pages pollute the retrieval index with low-quality passages that dilute your entity representation.

### Crawl-delay reality

As noted in chapter 3, `Crawl-delay` support is uneven. Googlebot and Applebot do not support it. Anthropic states that its bots support it where appropriate. The directive:

```
User-agent: GPTBot
Crawl-delay: 10
```

Requests a 10-second delay between requests. In practice, this is a polite suggestion, not a hard limit. For real rate limiting, use server-level controls (nginx `limit_req`, Cloudflare rate rules).

### robots.txt vs X-Robots-Tag

robots.txt controls whether a crawler can fetch a page. X-Robots-Tag (an HTTP header) controls whether a fetched page should be indexed. They operate at different levels:

- **robots.txt**: before fetch. The crawler checks robots.txt before requesting the page. If disallowed, it doesn't fetch.
- **X-Robots-Tag**: after fetch. The crawler receives the page, then checks the header. If `noindex`, it may fetch but won't index.

For AI crawlers, X-Robots-Tag support is inconsistent. Googlebot and Applebot document support for indexing directives; other AI crawlers and user-requested fetchers vary. The safest approach is to use robots.txt for crawl access control, and use meta robots or X-Robots-Tag only for indexing, snippet, or product-specific controls where the engine documents support. Do not rely on X-Robots-Tag as your primary AI access-control mechanism.

### robots.txt for large sites

For sites with thousands of pages, robots.txt isn't just about allow/disallow. It's about directing crawlers to the right content and away from the wrong content.

Best practices:

- **Disallow** search result pages, faceted/sort URLs, session URLs, cart/checkout pages, user account pages.
- **Allow** all public content pages: articles, products, documentation, about pages, category pages.
- **Reference your sitemap** in robots.txt:

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

This helps AI crawlers discover your content faster, especially for new or recently updated pages.

## Rendering: what the crawler actually sees

This is the most critical and most misunderstood access issue in GEO.

### The two rendering modes

When any crawler or fetcher requests a URL, it performs an HTTP GET request and receives the response body. What it does with that body depends on its rendering capability, product role, and whether the request is automatic crawl or user-initiated fetch:

**Basic fetch (no rendering)**: The crawler receives the raw HTML returned by the server. If the HTML contains the full content - text, headings, paragraphs, data - the crawler sees everything. If the HTML contains a JavaScript bundle and an empty `<div id="root">`, the crawler sees nothing.

**Rendered fetch**: The crawler executes JavaScript using a headless browser (Chromium, typically) and captures the DOM after JavaScript has run. This is what Googlebot does for traditional search - it renders pages in a two-phase process (initial fetch, then rendering queue).

Here's the critical question for GEO: **do AI crawlers render JavaScript?**

The answer, as of July 2026, is nuanced and unsatisfying:

- **Googlebot**: Renders JavaScript through Google's rendering system, but rendering can be delayed. For AI Overviews and fast discovery, initial HTML still matters.
- **Applebot**: Apple documents that Applebot may render content in a browser. Do not depend on rendering as your only access path; serve clean HTML anyway.

- **OAI-SearchBot**: Evidence is inconsistent. Some fetches appear capable of more than a raw fetch; others behave like basic HTML retrieval. Do not assume rendering.
- **PerplexityBot**: Treat as basic HTML retrieval for GEO planning. Content not in the initial HTML is at risk.
- **ClaudeBot / Claude-SearchBot / Claude-User**: Treat as basic HTML retrieval unless Anthropic documents a rendering capability for the surface you care about.
- **GPTBot**: Treat as basic fetch for training collection.

The practical conclusion: **assume AI crawlers do not render JavaScript.** If your content is not in the initial HTML response - the raw HTML the server returns before any JavaScript executes - it is invisible to most AI crawlers.

### What "not in the initial HTML" means

Let's be precise. When a user visits a client-side rendered (CSR) site - a React app, a Vue app, an Astro site with client-side hydration - their browser:

1. Downloads the HTML (which contains a `<div id="root"></div>` and `<script src="/bundle.js">`)
2. Downloads and executes the JavaScript bundle
3. The JavaScript renders content into the DOM
4. The user sees the page

Step 4 is what the user sees. Step 1 is what the crawler sees. If step 1 contains no content, the crawler sees nothing.

### Static HTML vs SSR vs SSG vs ISR

The rendering strategies, ranked by AI crawlability:

**Static HTML (SSG - Static Site Generation)**: The server pre-builds HTML files at build time. Every page is a complete HTML file with all content. The crawler receives the full content on the first request. This is the most AI-crawlable approach. Tools: Astro (static mode), Next.js (static export), Hugo, Jekyll, Eleventy, 11ty.

**Server-side rendering (SSR)**: The server renders HTML on each request using a template engine or framework. The crawler receives the full content on the first request. This is equally AI-crawlable. Tools: Next.js (SSR mode), Nuxt (SSR mode), Django templates, Laravel, Express with a template engine.

**Incremental Static Regeneration (ISR)**: Pages are statically generated at build time and revalidated at intervals. The first request after a revalidation triggers a regeneration. The crawler receives the static version (full content) in most cases. This is AI-crawlable. Tools: Next.js (ISR).

**Client-side rendering (CSR)**: The server sends a JavaScript shell. Content is rendered in the browser. The crawler receives an empty page. This is NOT AI-crawlable. Tools: React SPA, Vue SPA, Next.js with `output: 'export'` and client-only rendering.

**Hybrid**: Some pages are static, some are CSR. Common in Next.js App Router where some pages are server components and others are client components. Audit each page individually.

## How to check what the crawler sees

You don't need to guess. You can see exactly what a basic fetch returns.

**Method 1: curl**

```
curl -s https://yoursite.com/your-page | grep -i "your main heading text"
```

If the grep returns nothing, your heading text is not in the initial HTML. It's rendered by JavaScript, and AI crawlers likely can't see it.

**Method 2: View page source (not Dev Tools Elements)**

In your browser, right-click > "View Page Source" (not "Inspect Element"). Page Source shows the initial HTML. Inspect Element shows the rendered DOM. If your content appears in Inspect but not in Page Source, it's JavaScript-rendered.

**Method 3: Google Search Console URL Inspection**

Use the "Test Live URL" feature, then "View Tested Page" > "HTML" tab. This shows what Googlebot's initial fetch sees. If content is missing, it's not in the initial HTML.

**Method 4: GeoReady audit**

The rendering check in a GeoReady audit fetches your page with a basic HTTP client (no JavaScript execution) and checks whether the main content is present in the response.

## Fixing rendering issues

If your content is not in the initial HTML, you have three options, in order of preference:

**Option 1: Switch to SSG or SSR (best)**

If your site is a React/Vue SPA, migrate to a framework that supports server-side rendering or static generation. Next.js, Nuxt, Astro, SvelteKit, and Remix all support SSR/SSG out of the box.

This is the most reliable fix. It ensures every crawler - AI and traditional - sees your content immediately, without relying on JavaScript execution.

**Option 2: Pre-rendering (good)**

If a full migration isn't feasible, use pre-rendering: a service that generates static HTML snapshots of your pages and serves them to crawlers.

- **Prerender.io**: commercial service, detects crawler user-agents and serves pre-rendered HTML.
- **Rendertron**: open-source, self-hosted.
- **next-sitemap / react-snap**: build-time pre-rendering for React apps.

The risk: pre-rendering relies on user-agent detection. If a new AI crawler appears with an unknown user-agent, it may receive the JavaScript shell instead of the pre-rendered HTML. You need to keep your crawler list updated.

**Option 3: Hybrid rendering (acceptable)**

If some pages must be CSR (e.g., a dashboard behind login), ensure that all public-facing, content-bearing pages - the pages you want AI engines to cite - are SSG or SSR. Keep CSR only for authenticated, interactive areas that don't need AI visibility.

### The Astro advantage

Astro (the framework this site is built on) has a natural advantage for GEO. By default, Astro generates static HTML with zero JavaScript. Components are rendered at build time, and JavaScript is opt-in (via `client:` directives). This means:

- The initial HTML contains all content.
- AI crawlers see everything on the first fetch.
- No rendering queue, no JavaScript execution, no delay.

If you're choosing a framework for a content site where AI visibility matters, Astro is the strongest default. Next.js with SSG/SSR is equally good. Avoid CSR-only frameworks for content you want cited.

## Server responses: the HTTP status codes that matter

AI crawlers, like traditional search crawlers, interpret HTTP status codes to determine what to do with a page.

### 200 OK

The page exists and content was returned. The crawler indexes the content. This is what every page should return.

### 301 Moved Permanently

The page has a new permanent URL. The crawler follows the redirect and indexes the content at the new URL. 301s pass authority signals to the destination.

Use 301 for: domain changes, HTTP to HTTPS, URL restructuring, trailing slash normalization, www vs non-www.

**For AI crawlers**: 301s are followed and the destination is indexed. Ensure redirect chains are short (1-2 hops max) and the final destination returns 200.

### 302 Found / 307 Temporary Redirect

The page is temporarily at a different URL. The crawler may follow the redirect but typically indexes the original URL, not the destination.

**For AI crawlers**: treat 302/307 as temporary. If the redirect is actually permanent, use 301. Using 302 for permanent redirects causes AI crawlers to index the old URL, which may return a redirect on every fetch - wasting crawl budget and delaying content availability.

### 404 Not Found

The page doesn't exist. The crawler will stop trying to index it and may remove it from the index.

**For AI crawlers**: 404s are expected for genuinely removed pages. But if a page that was previously cited returns 404, the citation may disappear. Monitor for unintended 404s - especially after site migrations or CMS changes.

### 410 Gone

The page was permanently removed. More definitive than 404. The crawler treats this as "intentionally removed, don't retry."

**For AI crawlers**: use 410 for content you've deliberately removed and don't want indexed. It's faster than 404 for de-indexing.

### 403 Forbidden

The server refuses to return the page. The crawler interprets this as "access denied."

**For AI crawlers**: 403 is the most common silent killer of AI visibility. It means your server or a middleware (Cloudflare, WAF, bot protection) is blocking the crawler even though robots.txt allows it. If your server logs show AI crawlers receiving 403, your content is invisible to them - regardless of robots.txt configuration.

Common causes of 403 for AI crawlers: - Cloudflare bot fighting mode (challenges or blocks non-browser user-agents) - WAF rules that block unknown user-agents - Server-side bot detection that requires JavaScript challenge completion - Rate limiting that returns 403 instead of 429 - IP-based blocking (geo-blocks, VPN blocks)

### 429 Too Many Requests

The crawler is rate-limited. The crawler will back off and retry later.

**For AI crawlers**: occasional 429s are fine - the crawler retries. Persistent 429s (every request returns 429) effectively block the crawler. If your server rate-limits AI crawlers, ensure the limits are generous enough to allow reasonable crawl depth.

### 500 / 502 / 503 / 504

Server errors. The crawler will retry with exponential backoff. Persistent errors cause the crawler to stop visiting.

**For AI crawlers**: transient 5xx errors are normal. Persistent 5xx errors on specific pages cause those pages to be dropped from the index. Monitor your server error rate and fix 5xx-producing pages promptly.

### The status code audit

Check what AI crawlers actually receive:

```
# Simulate a basic fetch (no rendering, no cookies, no JS)
curl -s -o /dev/null -w "%{http_code}" -A "OAI-SearchBot" https://yoursite.com/your-page
curl -s -o /dev/null -w "%{http_code}" -A "PerplexityBot" https://yoursite.com/your-page
curl -s -o /dev/null -w "%{http_code}" -A "Claude-SearchBot" https://yoursite.com/your-
page
curl -s -o /dev/null -w "%{http_code}" -A "Googlebot" https://yoursite.com/your-page
curl -s -o /dev/null -w "%{http_code}" -A "Applebot" https://yoursite.com/your-page
curl -s -o /dev/null -w "%{http_code}" -A "GPTBot" https://yoursite.com/your-page
```

If any return non-200, investigate. The most common finding: Cloudflare or a WAF returning 403 to AI crawler user-agents.

## Middleware and obstacle layers

Between the crawler's HTTP request and your content, several layers can intercept and block or degrade the response.

### Cloudflare and CDN bot protection

Cloudflare's "Bot Fight Mode" and "Super Bot Fight Mode" are common causes of AI crawler 403s in audits. These features can challenge or block non-browser and bot-like user-agents, including GPTBot, OAI-SearchBot, PerplexityBot, Claude-SearchBot, Claude-User, and Applebot.

The irony: site owners enable Bot Fight Mode to prevent scraping, not realizing it also blocks the AI crawlers whose citations they want.

**The fix**: in Cloudflare, create WAF rules that explicitly allow AI crawler user-agents:

```
(http.user_agent contains "GPTBot") or (http.user_agent contains "OAI-SearchBot") or
(http.user_agent contains "ChatGPT-User") or (http.user_agent contains "PerplexityBot")
or (http.user_agent contains "Perplexity-User") or (http.user_agent contains
"ClaudeBot") or (http.user_agent contains "Claude-SearchBot") or (http.user_agent
contains "Claude-User") or (http.user_agent contains "Googlebot") or (http.user_agent
contains "Applebot")
→ Action: Allow
```

Place this rule above any bot-fighting rules. Verify by checking server logs for 200 responses from AI crawlers after the rule is deployed.

### WAF (Web Application Firewall) rules

Custom WAF rules that block unknown user-agents, non-browser headers, or requests without a `Referer` header will block AI crawlers. Review your WAF rules for:

- User-agent allowlists that only include browser user-agents
- Rules that block requests with empty `Accept-Language` headers (some AI crawlers don't send this)
- Rules that block requests without `Referer` or `Origin` headers
- Geo-blocking rules that accidentally block AI crawler IP ranges

### Cookie consent and consent banners

Cookie consent banners (CMPs - Consent Management Platforms) can block AI crawler access in two ways:

1. **JavaScript-rendered banners**: if the banner is JavaScript and the content is hidden behind it (display: none until consent), a non-rendering crawler may see the banner HTML but not the content - or may see both, but the banner HTML pollutes the content.
2. **Server-side consent gating**: some CMPs gate content server-side, requiring a consent cookie before serving the full page. AI crawlers don't have consent cookies, so they receive a reduced page or a consent wall.

**The fix**: ensure your content is in the initial HTML regardless of consent state. The cookie banner should be an overlay (visual), not a content gate. The content should be present in the DOM even if the banner is visible. AI crawlers should receive the full content without needing to accept cookies.

### Paywalls and login walls

If your content is behind a paywall or login, AI crawlers can't see it. This is by design - but it means paywalled content cannot be cited by AI engines.

Options:

- **Metered paywall**: allow AI crawlers to see the full content (first N articles free), and enforce the metering only for human visitors via JavaScript. The crawler receives the full

HTML; the paywall is a client-side overlay.

- **First-paragraph free**: include the first paragraph or summary in the initial HTML (accessible to crawlers), and gate the rest behind login. AI crawlers can cite the summary, even if they can't cite the full article.
- **Structured data for paywalled content**: use `isAccessibleForFree: false` and `hasCSSSelector` in JSON-LD to indicate paywalled sections. Google supports this for AI Overviews - it may cite the free portion.
- **Allow specific search crawlers and fetchers**: some publishers allow OAI-SearchBot, PerplexityBot, Claude-SearchBot, Claude-User, Googlebot, and Applebot while blocking GPTBot, ClaudeBot, Google-Extended, Applebot-Extended, and CCBot. This preserves citation eligibility while reducing training-data exposure where controls exist.

### Interstitials and pop-ups

Full-screen interstitials, pop-ups, and overlay ads don't block the HTML content - the content is still in the initial response. But they can:

- Push content below the fold, making it harder for crawlers to identify the main content
- Insert promotional text that gets embedded as a passage (replacing your actual content in the retrieval index)
- Create duplicate content signals if the interstitial text appears on every page

**The fix**: place interstitials at the end of the HTML, after the main content. Use CSS positioning (not DOM order) to display them as overlays. The DOM order should be: main content first, interstitials last.

## HTTP headers that influence AI crawling

Beyond robots.txt, several HTTP headers affect how AI crawlers process your content.

### X-Robots-Tag

```
X-Robots-Tag: noindex, nofollow
```

Tells crawlers not to index the page and not to follow links. Googlebot respects this. AI crawlers' support is inconsistent - treat it as a hint, not a rule.

Use cases: - `noindex` on search result pages, filter/sort URLs, and other low-value pages - `nofollow` on links to untrusted content - `noindex, follow` on pages you want crawled (for link discovery) but not indexed

### Content-Type

```
Content-Type: text/html; charset=utf-8
```

Ensures the crawler interprets the response as HTML. Missing or incorrect Content-Type headers (e.g., `application/octet-stream`) can cause crawlers to skip the page or misparse the content.

### Cache-Control

```
Cache-Control: public, max-age=3600
```

Controls how long the content can be cached. For AI crawlers, cacheable content reduces server load and speeds up re-crawling. Use `public` for content that should be cached by CDN and intermediaries.

Avoid `no-cache` or `no-store` for public content - it forces every request to hit your server and can slow crawl frequency.

### Vary

```
Vary: Accept-Encoding
```

Tells caches to serve different responses based on the request header. If your server serves different HTML to different user-agents (e.g., pre-rendered HTML for crawlers, CSR for browsers), use:

```
Vary: User-Agent
```

This is a signal to CDN and intermediary caches that the response varies by user-agent. Without it, a cached browser response might be served to a crawler (or vice versa).

**Warning**: `Vary: User-Agent` can significantly reduce cache hit rates. Only use it if you actually serve different content per user-agent.

### Link header

```
Link: <https://yoursite.com/canonical-page>; rel="canonical"
```

Specifies the canonical URL in the HTTP header. Useful for non-HTML resources (PDFs, images) where you can't add a `<link rel="canonical">` tag. Googlebot respects this. AI crawlers may not - but it's a good practice regardless.

## Site speed and crawl accessibility

AI crawlers, like traditional search crawlers, have a crawl budget - a limit on how many pages they'll fetch per site per time period. Slow pages consume more of that budget.

### Server response time (TTFB)

Time to First Byte (TTFB) is the time between the crawler's request and the first byte of the response. Target: under 200ms for most pages, under 500ms for all pages.

Slow TTFB causes: - Reduced crawl depth (the crawler gives up before reaching deep pages) - Reduced crawl frequency (the crawler visits less often) - Timeouts (the crawler abandons the request entirely)

Common causes of slow TTFB: - Database queries on every request (switch to cached/static where possible) - Heavy server-side rendering (cache rendered pages) - No CDN (all requests hit your origin server) - Shared hosting with limited resources

### Page weight

Total page weight (HTML + CSS + JS + images) affects crawl speed. AI crawlers typically download the HTML only - not CSS, JS, or images - so HTML weight matters most. Keep HTML under 200KB for content pages. Bloated HTML (inline styles, encoded images, unnecessary metadata) slows parsing and may cause the crawler to truncate content.

### Internal linking

Crawlers discover pages by following links. If your important pages are 5+ clicks from the homepage, AI crawlers may never reach them. Ensure:

- Every important page is within 3 clicks of the homepage
- Your sitemap lists all public pages (chapter 5 covers sitemap strategy)
- Internal links use real `<a href>` tags, not JavaScript onclick handlers (crawlers may not execute JS)

### Orphan pages

Pages with no internal links pointing to them are "orphan pages." Crawlers can only find them through the sitemap or external links. If a page is only in the sitemap and has no internal links, it may be crawled but will have weak entity association (because the crawler can't see how it relates to other pages on your site).

Fix orphan pages by adding contextual internal links from relevant existing pages.

## The access audit checklist

Run through this checklist to verify your access layer:

1. **robots.txt**: visit /robots.txt. Confirm AI search crawlers (OAI-SearchBot, PerplexityBot, Claude-SearchBot, Googlebot, Applebot) are allowed. Confirm user fetchers (ChatGPT-User, Perplexity-User, Claude-User) and training or AI-usage controls (GPTBot, ClaudeBot, Google-Extended, Applebot-Extended, CCBot) are intentionally allowed or blocked.
2. **Server logs**: grep for AI crawler and fetcher user-agents. Confirm 200 responses for OAI-SearchBot, PerplexityBot, Claude-SearchBot, Googlebot, and Applebot. If you see 403, investigate CDN/WAF rules and IP verification.
3. **Rendering**: curl your most important pages. Confirm the main content (headings, paragraphs, data) is in the initial HTML. If not, switch to SSG/SSR or add pre-rendering.
4. **HTTP status codes**: curl with AI crawler user-agents. Confirm 200 for all public content pages. Check for unintended 404s, 403s, or redirect chains.
5. **CDN/WAF**: check Cloudflare bot settings, WAF rules, and any bot protection. Ensure AI crawler user-agents are explicitly allowed.
6. **Cookie consent**: confirm content is in the initial HTML regardless of consent state. The banner should be a visual overlay, not a content gate.
7. **Paywalls**: if applicable, ensure AI crawlers can see at least the free portion of content. Use structured data to mark paywalled sections.
8. **Sitemap**: confirm sitemap.xml is referenced in robots.txt and lists all public content pages.
9. **Internal linking**: confirm every important page is within 3 clicks of the homepage. Check for orphan pages.
10. **TTFB**: measure server response time. Target under 200ms for key pages.

If all ten pass, your access layer is solid. If any fail, fix it before investing in schema, content, or entity signals. Access is the foundation - everything else builds on it.

---

*This is an excerpt from The GEO Readiness Manual. The full 160-page manual continues with Chapter 7 (Quotability) in this preview.*

---

# Chapter 7: Quotability - How to Write for Citation

## What quotability means

Quotability is the fourth and final layer of the GEO model. If access ensures the crawler can reach your content, orientation helps it find the right pages, and understanding lets it classify what the content is - quotability determines whether the AI can extract a self-contained, citable passage from your content.

This is not about writing style. It's about mechanics.

When an AI answer engine retrieves passages from its vector index, it looks for chunks of text that can be lifted from the page and inserted into a synthesized answer with minimal modification. A passage that is self-contained, direct, and unambiguous is quotable. A passage that requires context, interpretation, or restructuring is not.

The difference between quotable and non-quotable content is the difference between being cited and being skipped - even when your page is retrieved, even when your schema is valid, even when your entity is strong. If the model can't extract a clean passage, it will use a passage from another source that it can extract.

This chapter is about how to make your content mechanically extractable.

## How models select passages

To understand quotability, you need to understand what happens at the synthesis stage (chapter 2, stage 4).

The model receives a user question and a set of retrieved passages. It must construct an answer by synthesizing information from those passages. When it uses a passage, it cites the source. When it doesn't use a passage, it doesn't cite it.

The model doesn't rewrite passages extensively. It prefers to use passages with light editing - sometimes verbatim, sometimes with minor restructuring. This is why the format and style of your passages directly affects whether they get used.

Factors that influence passage selection:

1. **Self-containment**: can the passage be understood without reading the surrounding text? If a passage says "the third blocker is schema," the model can't use it without the preceding context (what are the first two blockers?). If it says "Missing or invalid JSON-LD schema is the third most common reason sites are invisible to AI engines," the passage is self-contained.

2. **Directness**: does the passage state the answer, or does it build toward it? A passage that opens with the answer and then elaborates is more likely to be extracted than one that builds to the answer over several sentences.
3. **Length**: passages of 40-120 words are ideal. Shorter passages may lack sufficient information. Longer passages may be truncated or skipped in favor of more concise alternatives.
4. **Factual precision**: passages with specific numbers, dates, names, and concrete claims are preferred over vague generalizations. "GeoReady's private benchmark contains 1000+ audit events, and the June 2026 public report covers 288 unique domains" is more quotable than "we've analyzed many sites."
5. **Non-redundancy**: if multiple sources say the same thing, the model uses one and cites one. Being first with a unique claim or unique data is more valuable than repeating a common claim.
6. **Structure**: passages with clear structure - definition, then explanation, then example - are easier to extract than unstructured prose. Lists, tables, and numbered items are highly extractable.

## The passage extraction model

Here's a concrete way to think about it. Imagine your page is split into 200-500 token chunks by the embedding model. Each chunk becomes an independent passage in the vector index. When a user asks a question, the chunks closest to the question are retrieved.

Now ask: which chunks from your page would be retrieved for a relevant question? And of those, which are quotable?

Take this paragraph:

> *In today's rapidly evolving digital landscape, businesses face an unprecedented challenge in maintaining visibility across an increasingly fragmented array of platforms. As artificial intelligence continues to reshape how users discover and consume information, it has become clear that traditional approaches to search engine optimization, while still relevant, may no longer be sufficient to ensure that your content reaches its intended audience.*

This is 65 words. It contains zero quotable claims. No specific fact, no direct answer, no actionable statement. An AI engine would never extract this passage because there's nothing to extract. It's atmosphere, not information.

Now take this paragraph:

> *When a site isn't cited by ChatGPT, the problem is almost never content quality. It's infrastructure: robots.txt blocking GPTBot, JavaScript-only rendering, missing JSON-LD schema, non-extractable content, or weak entity identity. Each of these takes 5 minutes to diagnose and an afternoon to fix.*

This is 45 words. It contains a quotable claim ("the problem is almost never content - it's infrastructure"), a list of five specific causes, and a concrete time estimate. An AI engine can extract this almost verbatim and cite it.

The difference is not vocabulary or tone. It's information density and self-containment.

## Writing patterns that get cited

### Pattern 1: BLUF - Bottom Line Up Front

State the answer in the first sentence. Elaborate after.

**Not quotable:**

> There are many factors that contribute to a website's visibility in AI-powered search results. Over the course of our research, we've identified several key areas that tend to cause problems for site owners. One of the most significant is the configuration of the robots.txt file, which can inadvertently block access to AI crawlers.

**Quotable:**

> The most common reason a site is invisible to AI engines is a robots.txt file that blocks GPTBot or PerplexityBot. The fix is usually one line. Check your /robots.txt for Disallow directives targeting AI crawler user-agents.

The second version leads with the answer. The model can extract the first sentence and cite it. The first version builds to the answer over three sentences - the model would need to synthesize across all three, and the first two sentences add no quotable content.

### Pattern 2: Definition + explanation + example

Define a concept, explain it, give a concrete example. Each part is independently quotable.

**Quotable:**

> llms.txt is a Markdown file at your domain root that tells AI tools which pages matter and what they're about. Think of it as a sitemap for AI: instead of listing URLs, it explains what each section contains and why it matters. For example, a SaaS site might have sections for features, pricing, documentation, and API reference - each with a one-sentence description.

The first sentence is a definition (quotable for "what is llms.txt"). The second is an analogy (quotable for "how does llms.txt work"). The third is a concrete example (quotable for "llms.txt example").

### Pattern 3: Numbered lists with self-contained items

Each list item should be understandable without reading the other items.

**Not quotable:**

> 1. The first issue is robots.txt.
> 2. The second is JavaScript rendering.
> 3. The third is schema.

These items require the surrounding context ("the five blockers are...") to make sense. The model can't extract item 2 alone.

**Quotable:**

> 1. robots.txt blocking GPTBot or PerplexityBot - often inherited from outdated anti-scraping rules. The site is invisible by default.
> 2. JavaScript-only rendering. If content requires React or Vue to appear, a basic AI crawler fetch sees nothing. Static HTML or SSR isn't optional for GEO.
> 3. Missing or invalid JSON-LD schema. Without structured data, the AI must infer what you are. Inference means errors.

Each item is self-contained. The model can extract any one of them independently and cite it.

### Pattern 4: Tables and structured data

Tables are highly extractable because they present information in a structured, unambiguous format. AI engines can extract entire rows or columns.

**Quotable:**

| SEO Signal | GEO Equivalent | Why It Changed |
|---|---|---|
| Keywords | Entities | Models map topics, not keyword density |
| Backlinks | Unlinked mentions | Authority comes from being referenced, not linked |
| Meta description | Quotable passages | AI extracts body text, not meta tags |
| Page authority | Entity authority | Clusters of pages beat single strong pages |
| Click-through rate | Citation rate | You measure whether the model names you |

A table like this can be extracted almost verbatim. The structure IS the content - no interpretation needed.

### Pattern 5: Direct answers to direct questions

If your content answers a question, state the answer first, then qualify it.

**Quotable:**

> How long does it take to get cited by ChatGPT? There is no fixed timeline, but three factors control most of the variance: crawl frequency, content structure, and entity saturation. If GPTBot visits weekly and your content answers the question in the first paragraph, new content can be cited within days. If your site is unknown to the model, it takes longer - the model needs to build confidence first.

The first sentence answers the question directly. The rest elaborates with specifics. The model can extract the first sentence, the first two sentences, or the entire paragraph - all are self-contained at different levels of detail.

## Writing patterns that don't get cited

### Anti-pattern 1: The throat-clearing introduction

> In an era where artificial intelligence is fundamentally transforming the digital landscape, it is imperative for organizations to critically evaluate their approach to online visibility. This comprehensive analysis delves into the multifaceted considerations that...

No quotable content. The model skips this. Yet many articles open with 2-3 paragraphs of throat-clearing before reaching the first quotable claim. Those 2-3 paragraphs are wasted - they're crawled, indexed, and embedded, but never retrieved because they're semantically distant from any useful query.

**Fix**: delete the introduction. Start with the answer. If context is needed, provide it in one sentence, then move to the answer.

### Anti-pattern 2: The hedge

> While it could be argued that robots.txt may play a somewhat significant role in potentially limiting AI crawler access to certain pages, it's important to note that this might not always be the case, and results may vary depending on individual circumstances.

The claim is buried under five layers of hedging. The model can't extract a clear statement. Compare:

> robots.txt is the single most common reason a site is invisible to AI engines. The fix is usually one line.

Same claim, stated directly. Quotable.

**Fix**: state claims directly. If you're uncertain, attribute them: "in GeoReady's private benchmark," "in the June 2026 State of GEO report," or "in 1000+ private audit events." That is attribution, not hedging. Hedging is "might possibly perhaps somewhat." Attribution is "we observed this in a defined benchmark." The former dilutes; the latter strengthens.

### Anti-pattern 3: The narrative journey

> Let me take you on a journey through the world of GEO. It all started when I first noticed that my site wasn't appearing in ChatGPT answers. I began researching, and after months of experimentation, I discovered that the problem was simpler than I thought...

Narrative writing is engaging for humans but hostile to passage extraction. The quotable content is buried inside a personal story, and the model can't extract the claim without also extracting the narrative frame.

**Fix**: separate the story from the claims. If you want a narrative hook, keep it to one sentence, then switch to direct, quotable statements for the substantive content.

### Anti-pattern 4: The link-dependent paragraph

> As we discussed in the previous section, the third factor is also related to the issues mentioned earlier. For more details, see our guide on robots.txt configuration, which covers this topic in depth.

This paragraph is not self-contained. It references other sections, other pages, and provides no standalone content. The model can't extract anything useful from it.

**Fix**: each paragraph should make a complete point without requiring the reader to have read previous paragraphs or follow links. Cross-references are fine, but they should supplement, not replace, the content.

### Anti-pattern 5: The unstructured wall of text

A 500-word paragraph with no headings, no lists, no structure. The embedding model chunks it into 200-500 token passages, but none of the chunks have clear boundaries. The model may extract a chunk that starts mid-sentence and ends mid-sentence, producing an incoherent passage.

**Fix**: break long paragraphs into 2-4 sentence blocks. Each block should make one point. Use headings to separate sections. Use lists for enumerations. Structure makes chunks cleaner.

## The quotability audit: before and after

Here's a real example from a GeoReady audit, rewritten for quotability.

**Before (not quotable):**

> When it comes to ensuring your website is properly optimized for AI-powered search engines, there are many different factors that need to be taken into consideration. One of the most important things to keep in mind is that your content needs to be structured in a way that makes it easy for AI systems to understand and process. This means using appropriate formatting, including headers and subheaders, and making sure that your main points are clearly articulated and easy to find within the text of your pages. Additionally, it's worth noting that the use of structured data can also play a significant role in helping AI engines better comprehend the nature and purpose of your content.

87 words. Zero quotable claims. The entire paragraph could be deleted without losing any information.

**After (quotable):**

> AI engines extract content in passages of 40-120 words. To be cited, each passage must be self-contained: state the claim first, then elaborate. Use headings, lists, and tables - structured content is easier to extract than prose. Add JSON-LD schema so the AI can classify the page without inferring from text.

47 words. Four quotable claims, each extractable independently.

## Headings as retrieval signals

Headings serve a dual purpose in GEO: they structure the page for human readers, and they provide semantic context for AI passage extraction.

When the embedding model chunks your page, it uses headings as boundary signals. A passage that starts at an H2 is more likely to be a clean, self-contained chunk than a passage that starts in the middle of a paragraph under no heading.

Best practices for headings:

- **Use question-format headings for FAQ-style content**: "How long does it take to get cited by ChatGPT?" is a better heading than "Timeline for AI citation." The question format matches user queries and improves semantic retrieval.
- **Keep headings descriptive, not clever**: "robots.txt: the single most common AI visibility blocker" is better than "The Gatekeeper Problem." Clever headings may engage humans but confuse semantic retrieval.
- **Use H2 for major sections, H3 for subsections**: maintain a clear hierarchy. Don't skip levels (H2 to H4). The hierarchy helps the model understand the page structure.
- **Include the key term in the heading**: if the section is about llms.txt, the heading should include "llms.txt." This improves semantic matching between the heading, the passage, and the user query.
- **One H1 per page**: the H1 should match the Article schema headline (chapter 6). Multiple H1s confuse the page structure signal.

## Numbers, dates, and proper nouns

AI engines prefer passages with concrete, verifiable details. A passage that says "GeoReady's private benchmark contains 1000+ audit events, and the June 2026 public State of GEO report covers 288 unique domains with an average score of 53.6/100" is more quotable than one that says "many sites have AI crawler access issues."

Specific signals that increase quotability:

- **Numbers**: "1000+ private audit events," "288 unique domains," "53.6/100 average score," "3-7 sources per answer," "5-minute test"
- **Dates**: "as of July 2026," "published July 12, 2026"

- **Proper nouns**: "GPTBot," "OAI-SearchBot," "PerplexityBot," "Schema.org"
- **Concrete claims**: "the fix is usually one line," "static HTML or SSR isn't optional for GEO"
- **Named methodologies**: "the four-layer model: access, orientation, understanding, quotability"

These signals serve two purposes: they make the passage more informative (higher value to the model), and they make the passage more semantically distinct (higher uniqueness in the vector space, reducing redundancy with other sources).

## Voice and tone for citation

The tone that gets cited is not the tone that goes viral. Viral content is often provocative, emotional, or contrarian. Cited content is precise, confident, and verifiable.

Principles:

- **Assert, don't suggest**: "robots.txt is the most common blocker" (assertion) vs "robots.txt might be a common blocker" (suggestion). Assertions get cited. Suggestions get skipped.
- **Attribute, don't hedge**: "in GeoReady's private benchmark of 1000+ audit events" or "in the June 2026 State of GEO report" (attribution) vs "it seems likely that" (hedging). Attribution strengthens. Hedging dilutes.
- **Be specific, not general**: "GPTBot, OAI-SearchBot, and PerplexityBot" (specific) vs "various AI crawlers" (general). Specifics are quotable. Generalities are not.
- **Be concise, not comprehensive**: a 45-word passage with one clear claim is more quotable than a 200-word passage that covers every aspect. Comprehensive content serves human readers; concise passages serve AI extraction. You can have both - write the concise passage first, then elaborate for humans.
- **Use technical terms precisely**: "vector embedding" not "AI representation," "retrieval-augmented generation" not "AI answer generation." Precise terminology improves semantic matching and signals expertise.

## The quotability checklist

For each page you want AI engines to cite, check:

1. **First paragraph**: does it state the main claim directly, or does it throat-clear? Rewrite to lead with the answer.
2. **Self-containment**: can each paragraph be understood without reading the previous one? If not, add context or restructure.
3. **Passage length**: are your paragraphs 40-120 words? Break longer paragraphs into shorter, self-contained blocks.

4. **Headings**: do they use question format for FAQ content? Do they include the key term? Is the hierarchy clean (H1 > H2 > H3)?
5. **Lists and tables**: are enumerations presented as numbered lists with self-contained items? Is comparative data in tables?
6. **Factual precision**: does every section include at least one specific number, date, or proper noun?
7. **Hedging scan**: search for "might," "could," "possibly," "perhaps," "it seems." Replace with direct assertions or attributions.
8. **Throat-clearing scan**: can you delete the first 1-2 paragraphs of each section without losing information? If yes, delete them.
9. **Link-dependency scan**: does any paragraph require following a link to make sense? Rewrite to be self-contained.
10. **Structure scan**: is the page a wall of text, or is it broken into headed sections with lists and tables? Add structure.

If all ten pass, your content is quotable. AI engines that retrieve your passages will find them easy to extract, and citation probability increases significantly.

## The relationship between quotability and content quality

Quotability is not a substitute for content quality. It's a structural layer on top of quality content. A page that is quotable but wrong, superficial, or unoriginal may be cited once but won't be cited again - the model learns from user feedback and engagement signals.

The goal is content that is both high-quality and quotable: accurate, original, well-researched content that is also structured for passage extraction. Quality gives the model a reason to cite you. Quotability gives the model the ability to cite you.

If you have quality but not quotability, the model wants to cite you but can't extract a clean passage. If you have quotability but not quality, the model can extract a passage but doesn't choose to because the content doesn't add value. Both are needed.

---



---