Barkhausen AI

NOTE

What robots.txt does and doesn't control for AI crawlers

Operators publishing a robots.txt file often intend to make one decision — keep this site out of AI training data — and instead make several, because AI-related crawlers are not one thing.

“Every leap begins as a crackle.”

KIND	Note
DATE	2026
SUBJECTS	Crawlers & access
LICENSE	CC-BY-4.0
AUTHOR	Barkhausen AI

ABSTRACT

Operators publishing a robots.txt file often intend to make one decision — keep this site out of AI training data — and instead make several, because AI-related crawlers are not one thing. The same operator, and often the same AI company, runs separate crawlers for training-data collection, for building a retrieval index, and for fetching a page in real time when a user asks about it, and each is controlled by a distinct token. Disallowing the training crawler does not disallow the others, and a rule written broadly enough to catch more than one class can remove a site from AI-generated answers as an unintended side effect. This note reads the public documentation for named crawler classes and states plainly what robots.txt is: a request, not an enforcement mechanism.

CONTENTS

A robots.txt file that disallows one AI crawler is often assumed to have opted a site out of "AI" broadly. That assumption fails because the crawlers named in a modern robots.txt file are not doing the same job. Public documentation from the operators involved describes at least three distinct functions — collecting content for model training, building a retrieval index used to answer search-style queries, and fetching a specific page in real time because a user asked about it — and each function is controlled by its own, separately named token. A rule aimed at one does not reach the others.

1 Three jobs, three tokens

OpenAI's own crawler documentation names this split directly for its own systems: GPTBot "is used to crawl content that may be used in training" OpenAI's generative AI foundation models; OAI-SearchBot "is used to surface websites in search results in ChatGPT's search features," a retrieval-indexing function distinct from training collection; and ChatGPT-User visits a page only "when users ask ChatGPT or a CustomGPT a question," a fetch OpenAI treats as user-initiated rather than automatic crawling [1]. The same three-way split recurs, in different form, across other operators. Google-Extended is not a separate crawler with its own user-agent string at all; Google's documentation describes it as "a standalone product token" applied within existing Google crawler identifiers, controlling only whether crawled content may be used "for training future generations of Gemini models" and for grounding those models' answers [2]. Common Crawl's CCBot is a general-purpose web crawler, unaffiliated with any single AI company, whose resulting open dataset is a widely used input into many organizations' training pipelines, and it is disallowed through its own distinct token in robots.txt [3].

2 Why blocking one does not block the others

Because each function is controlled by a separate token, a robots.txt rule has to name the specific token it intends to affect. Disallowing GPTBot signals a training opt-out and, per OpenAI's documentation, has no bearing on whether OAI-SearchBot may still crawl the same site for search surfacing [1]. The reverse holds too: disallowing OAI-SearchBot removes a site from ChatGPT's search-style answers without affecting whether GPTBot may still collect the site's content for training, since the two are governed independently. Google states the same independence explicitly for its own tokens: disallowing Google-Extended "does not impact a site's inclusion in Google Search nor is it used as a ranking signal in Google Search" [2] — the training-and-grounding control and the search-indexing crawler are documented as separate levers, not one switch.

3 The common misconfiguration

The practical consequence is a mis-block: an operator who intends only to keep a site out of model training, and writes a robots.txt rule broadly enough to also catch a retrieval-indexing crawler, can remove the site from AI-generated search answers as a side effect of a decision aimed only at training. The reverse mistake is equally possible — permitting broad access while believing a narrower, training-only exclusion was in place. Because the tokens are separate strings maintained separately by each operator and revised over time, a rule that was correct when written can become incorrect, or ambiguous, as new tokens are introduced. A current, maintained mapping of named tokens to the class of crawling they perform is kept at the AI crawler registry.

4 robots.txt is a request, not a lock

None of the distinctions above change what robots.txt fundamentally is. The IETF specification governing the file states plainly that "these rules are not a form of access authorization" [4]; a robots.txt entry is a request that a crawler is "requested to honor," not a barrier the file itself enforces. Whether any given crawler actually complies with a published robots.txt file is a decision made by whoever operates that crawler, not by the file.

5 Limitations

This note describes named public tokens as documented by three operators at a single point in time; token names, the crawlers behind them, and their documented scope change, and this note is not the maintained record of that — the linked registry page is. It also does not address the separate, harder-to-verify empirical question of whether a given crawler's behavior in practice matches its own documentation; that question requires independent measurement and is out of scope here.

REFERENCES

1. OpenAI. *Overview of OpenAI Crawlers*. https://developers.openai.com/api/docs/bots Accessed 2026-07-08. [archived]
2. Google, Google Search Central documentation. *Google's common crawlers*. https://developers.google.com/crawling/docs/crawlers-fetchers/google-common-crawlers Accessed 2026-07-09. [archived]
3. Common Crawl Foundation. *CCBot*. https://commoncrawl.org/ccbot Accessed 2026-07-08. [archived]
4. M. Koster, G. Illyes, H. Zeller, and L. Sassman, IETF. *RFC 9309: Robots Exclusion Protocol* (2022). https://www.rfc-editor.org/rfc/rfc9309.html Accessed 2026-07-08. [archived]

HOW TO CITE

Barkhausen AI (2026). What robots.txt does and doesn't control for AI crawlers. https://barkhausen.ai/notes/robots-txt-and-ai-crawlers/