Troubleshooting Guide: When Bots & Crawlers Can't Read a Website or llms.txt File

This troubleshooting document is provided as the steps that I took to troubleshoot and test the llms.txt file that I created for my website. It should be considered as technical advice, consult your web developer for qualified recommendations.

If you find value in what I shared on the podcast and in this free troubleshooting guide, I'd so appreciate you supporting my show with a listen and a star rating and review. You might find that it improves your health and become a regular listener, we'd love to welcome you to the community. New Normal Big Life streams free in audio and video on **Spotify** and **Apple Podcasts**.

Best,

Antoinette Berrafato

Let's connect!

X: https://x.com/NNBLBlog

IG: https://www.instagram.com/nnblpodcast/

TikTok: https://www.tiktok.com/@newnormalbiglifep

F: https://www.facebook.com/NewNormalBigLifePodcast/

In: https://www.linkedin.com/in/antoinetteberrafatomba/

YouTube: https://www.youtube.com/@NewNormalBigLifePodcast

Listen to New Normal Big Life wherever you get your podcasts

https://newnormalbiglife.buzzsprout.com/

W: https://newnormalbiglifepodcast.com

Step 1: Confirm the Problem Is Bot-Specific

Before digging in, pinpoint who can't access the file.

Open the URL in a normal browser (e.g., https://yoursite.com/llms.txt). If it loads fine here but tools/crawlers get nothing, the issue is almost certainly bot blocking, not a missing file.

Test in incognito/private mode to rule out caching or logged-in sessions fooling you.

Try a different network or device to eliminate local issues.

Step 2: Verify the File Actually Exists and Is Correct

Rule out simple placement and naming mistakes.

Location: The file should sit in the root directory -> https://yoursite.com/llms.txt.

Exact name: All lowercase, correct extension (llms.txt, not LLMs.txt or llms.txt.txt).

Content type: It should be served as text/plain.

Status code: A direct request should return 200 OK, not a 404 (missing), 403 (forbidden), or 5xx (server error).

Step 3: Check robots.txt

This file tells crawlers what they may or may not access.

Open https://yoursite.com/robots.txt.

Look for any Disallow rules that block /llms.txt or the whole site (e.g., Disallow: /).

Confirm the relevant user agents (e.g., User-agent: *) are actually allowed to crawl the paths you care about.

Step 4: Inspect Security Plugins & Firewalls

This is the most common culprit for "browser works, bots don't."

Security plugins (WordPress: Wordfence, Sucuri, iThemes) often auto-block non-browser traffic. Review their firewall and rate-limiting settings.

CDN / WAF layers (Cloudflare, Akamai, AWS WAF) frequently have "Bot Fight Mode," "AI Scraper Blocking," or challenge pages (CAPTCHA/JavaScript checks) that stop crawlers cold.

Whitelist the specific crawlers or user agents you want to allow.

Step 5: Look at User-Agent & Header Handling

Some servers respond differently based on how the request is made.

Servers may serve content to browser user-agent strings but block or blank out unknown/automated ones.

JavaScript-rendered content can appear in a browser but return empty to simple fetchers that don't run JS.

Confirm the server isn't requiring cookies, headers, or JS execution just to view a plain text file.

Step 6: Rule Out Temporary Issues

Not every problem is a permanent block.

Server outages or overload can return blank or error responses intermittently.

Aggressive caching (server-side or CDN) may serve a stale/empty version-try clearing the cache and re-testing.

Retry after a short wait to see if the issue was transient.

Step 7: Test With Diagnostic Tools

Confirm your fix from a non-browser perspective.

Use a command-line tool like curl or an online HTTP header checker to see the exact status code and content returned to a non-browser request.

Compare that response to what you see in the browser-if they differ, you've isolated the block.

Quick Reference Checklist

Check	**What You're Confirming**
Loads in browser but not for bots	Points to bot-specific blocking
File in root, lowercase, .txt	Correct placement & naming
Returns 200 (not 403/404/5xx)	File is accessible
robots.txt allows the path	Crawlers are permitted
Security plugin/CDN settings	No bot-fight or AI-scraper blocks
Served as text/plain, no JS required	Content is directly readable
Cache cleared, no outage	Not a temporary glitch

The fastest first move: if it opens in a browser but fails for bots, jump straight to Step 4 (security plugins/CDN)-that resolves the majority of these cases.

There is another reason why your llms.txt file is returning a 403 error that is not related to robots.txt, but rather to the current .htaccess rules on your hosting account. By default, only robots.txt and ads.txt are whitelisted, which means other .txt files are blocked.

To correct this, you'll need to update the rewrite condition in your .htaccess file. Please replace the following line:

RewriteCond %{REQUEST_URI} !(robots.txt|ads.txt|[a-z0-9_-]*sitemap[a-z0-9_.-]*.(xml|xsl|html)(.gz)?)

with:

RewriteCond %{REQUEST_URI} !(robots.txt|ads.txt|llms.txt|[a-z0-9_-]*sitemap[a-z0-9_.-]*.(xml|xsl|html)(.gz)?)

This change adds llms.txt to the whitelist, allowing it to be served directly by the server just like robots.txt.

After saving the update, you can test the file after 2-24 hours before running tests with AI bots, as caching and propagation may take some time.