FROM THE WORKBENCH · NOVEMBER 28, 2026

How to Serve Clean Markdown Twins and Link Headers on WordPress

By Will Tygart · Tygart Media · Canonical guide & .md twin at https://tygartmedia.com/how-to-serve-markdown-twins-link-headers-wordpress/

Direct Answer: To serve clean Markdown twins on WordPress for coding terminals, deploy LF UTF-8 <slug>.md files directly to your webroot alongside HTML posts and wire four discovery pointers on one canonical URL: RFC 8288 HTTP Link headers, HTML <head> alternate links, Schema.org encoding arrays, and visible page buttons.

When an engineer runs a prompt inside a terminal or IDE, their agent rarely wants fifty kilobytes of theme CSS, navigation menus, and inline scripts. Across our 31.8-day telemetry audit in Four AI Middlemen Out-Crawled Googlebot, a typical WordPress article page weighed 50,646 bytes uncompressed while its clean Markdown twin weighed just 8,515 bytes — a 5.9x payload reduction.

Wait until you check how different agents discover those files. In 3,405 Visits From Readers Without a Browser, CLI agents explicitly sent `Accept: text/markdown` headers and pulled 55 direct `.md` files cleanly. Meanwhile, bulk scrapers routinely ignored `<head>` tags unless visible links appeared in the HTML body or HTTP headers. Wiring all four rails onto the same canonical URL serves every client without splitting your SEO equity.

The Four Machine Discovery Rails on One Canonical URL

Never spin up separate per-bot directories or return cloaked HTML by User-Agent. Keep one canonical URL and expose the static `.md` twin uniformly across all four RFC 8288 and HTML pointers:

Discovery Rail	Protocol / Layer	Best Consumer Clients	Verification Requirement
1. HTTP Link Header	RFC 8288 Link header via PHP send_headers	CLI terminals and headless fetchers checking HEAD/GET	Header present on single posts; zero duplicate Link headers on homepage
2. HTML Head Alternate	link rel='alternate' type='text/markdown'	Standards-compliant crawlers and Markdown-aware browsers	Exact single tag count == 1 matching live HTTP 200 .md path
3. Schema.org encoding	Article JSON-LD MediaObject encodingFormat text/markdown	Structured knowledge graphs and schema extractors	Valid JSON-LD citing canonical contentUrl
4. Visible Body Button	Inline HTML anchor pointing directly to slug.md	Search APIs and scrapers that ignore head metadata	Verified 200 status and byte-exact SHA-256 against local source

For broader catalog indexing alongside per-post twins, also maintain an llms.txt index at your domain root — remembering never to overwrite a dynamic SEO plugin route with a static file if your stack already generates one dynamically.

WordPress MU-Plugin to Emit Both Link Headers and Head Tags

Block themes omit Markdown alternate links by default, and WordPress core REST media uploads reject `.md` files with `HTTP 500 rest_upload_sideload_error`. Deploy `.md` files directly to your webroot over SSH/SFTP, then drop a compact Must-Use plugin inside `wp-content/mu-plugins/` that checks disk existence before advertising the alternate URL:

wp-content/mu-plugins/site-markdown-alternate.php (No Global Vars Needed)

```
<?php
/* Plugin Name: Markdown Twin Discovery Rail */
if (!defined('ABSPATH')) { exit; }

function tm_resolve_md_twin_url(): string {
if (!is_singular('post') || is_front_page()) { return ''; }
if (!get_post_field('post_name', get_queried_object_id())) { return ''; }
if (!file_exists(ABSPATH . get_post_field('post_name', get_queried_object_id()) . '.md')) { return ''; }
return home_url('/' . get_post_field('post_name', get_queried_object_id()) . '.md');
}

add_action('wp_head', function (): void {
if (tm_resolve_md_twin_url() !== '') {
echo '<link rel="alternate" type="text/markdown" href="' . esc_url(tm_resolve_md_twin_url()) . '">' . "\n";
}
}, 1);

add_action('send_headers', function (): void {
if (tm_resolve_md_twin_url() !== '') {
header('Link: <' . esc_url_raw(tm_resolve_md_twin_url()) . '>; rel="alternate"; type="text/markdown"', false);
}
}, 20);
```

Nginx MIME Configuration, UTF-8 Decoding, and Common Traps

Markdown Twin Trap	Root Technical Cause	Concrete Symptom	Operational Fix
WP REST Media upload rejects .md	WordPress core blocks Markdown uploads via rest_upload_sideload_error	Deterministic HTTP 500 on POST /wp-json/wp/v2/media	Stage LF .md tarballs over SSH and install with web-user ownership
ISO-8859-1 smart-punctuation mojibake	Python requests defaults charset-less text/* types to ISO-8859-1	Em-dashes corrupt when reading resp.text instead of UTF-8 bytes	Set Content-Type charset=utf-8 in Nginx and decode resp.content UTF-8
CRLF vs LF SHA-256 drift	Windows editors write CRLF unless newline='\n' is forced	Disk file fails exact SHA-256 verification against Linux origin	Force newline='\n' on Python write_text and extract via tar -mxzf
Short italic byline before Direct Answer	Chunkers score p[0] under H1 as opening passage density	Sub-30-word byline tanks RAG readiness scores into low 50s	Place 45-55w Direct Answer paragraph immediately after H1 heading

Nginx Markdown MIME Block + Python UTF-8 Verification Check

```
# 1. Nginx server block (text/markdown; charset=utf-8)
# location ~* \.md {
# types { text/markdown md; }
# default_type text/markdown;
# charset utf-8;
# charset_types text/markdown text/plain;
# try_files /wp-content/uploads/md-cache =404;
# }
# (Simply point location at your static webroot file path with try_files uri =404)

# 2. Write explicit LF UTF-8 locally & decode strictly as UTF-8 on verify
import hashlib, pathlib, urllib.request

def publish_prepare_and_verify(md_text: str, out_path: pathlib.Path, live_url: str) -> None:
clean = md_text.replace("\r\n", "\n")
out_path.write_text(clean, encoding="utf-8", newline="\n")
expected_sha = hashlib.sha256(out_path.read_bytes()).hexdigest()
req = urllib.request.Request(live_url, headers={"User-Agent": "Mozilla/5.0"})
with urllib.request.urlopen(req, timeout=15) as r:
assert "markdown" in r.headers.get("Content-Type", "").lower()
live_bytes = r.read()
assert hashlib.sha256(live_bytes).hexdigest() == expected_sha
```

Make Every How-To Pullable From a Terminal

Remember RFC 9110 content and caching rules: clear any page cache or reverse-proxy cache layer right after updating `.md` files and warm key URLs with GET requests rather than HEAD probes. When an engineer asks their coding agent for help mid-debug, a clean 8.5 KB Markdown spec arrives inside their editor before a bloated HTML page finishes rendering.

> **Workbench Next Step:** Publishing a canonical Markdown twin beside WordPress HTML cuts bot and terminal transfer size 5.9x across four discovery rails. Continue at https://tygartmedia.com/how-to-wire-wordpress-mcp-grok-claude-session-teardown/