

Restoring a Single WordPress Page from a WP Engine Backup — Without Touching the Rest of the Site

How I used Claude AI to extract one page's data from a full database dump and push it back directly via the WordPress MCP — no REST API calls, no manual copy-paste, no downtime.

Pranoti Kshirsagar · AI Integration & Automation Specialist · thesciencetalk.com

1

page restored precisely

0

other pages affected

<10 min

vs. hours of full restore

01 · THE PROBLEM — WORDPRESS RESTORES EVERYTHING OR NOTHING

THE GOAL

Recover **one broken or deleted page** from an existing WP Engine backup — without disrupting the dozens of other pages, posts, and settings on the live site.

THE DAILY REALITY

WP Engine's standard restore replaces the **entire site with a snapshot**. Any content created after the backup date is wiped. There is no built-in "restore single page" option.

02 · BEFORE VS AFTER — THE WORKFLOW

Before

1
Notice broken page

>

2
Open WP Engine dashboard

>

3
Restore full site snapshot

>

4
All recent content wiped

>

5
Manually re-enter lost content

After

1
Download database dump from WP Engine backup

>

2
Feed SQL dump to Claude — extract target page data

>

3
Push page back via WordPress MCP — live in seconds

02B · THE ASSUMPTION THAT MAKES THIS FEEL IMPOSSIBLE

COMMON ASSUMPTION THAT'S WRONG

"You need to restore the whole backup to get one page back." In reality, the page content is sitting right there in the database dump as a plain **SQL INSERT statement**. You don't need to run the full restore — you just need to read the dump, extract the right row, and write it back using the WordPress MCP directly from Claude.

03 · BLOCKERS AND SOLUTIONS

BLOCKER

SOLUTION

✗ **SQL dump is massive and unreadable** — A full WP Engine database export is thousands of lines of SQL. Finding the right INSERT for one specific page by hand is impractical.

✓ **Claude reads the dump** and locates the exact row for the target page in the `wp_posts` table — by post ID, slug, or title — and returns just the content you need.

✗ **MCP needs to be connected and configured** — WordPress MCP only works if it's already set up and authenticated against the right site. Without it, Claude has no write access.

✓ **MCP was already connected** to `thesciencetalk.com`. Claude called `update_page` directly — no separate auth step, no manual API calls, no credential formatting needed.

✗ **Page ID mismatch between backup and live site** — If the page was deleted and re-created, its ID in the dump may differ from any existing draft on the live site.

✓ **Claude resolved the ID via MCP** — used `get_pages` to find the current page ID on the live site, then called `update_page` with the recovered content. No guesswork.

04 · WHAT BECAME POSSIBLE

✓ Target page fully recovered with original content, slug, and formatting intact

✓ Zero other pages, posts, or settings touched on the live site

✓ No downtime — the live site stayed online throughout the entire recovery

✓ Replicable workflow — same approach works for any page or post in any WP Engine backup

05 · HOW THE RECOVERY WORKFLOW IS STRUCTURED

ARCHITECTURE

- 1 **WP Engine backup dashboard** — Download the database dump (.sql file) from the relevant backup checkpoint via the WP Engine user portal.
- 2 **Claude AI (data extraction)** — Paste or upload the SQL dump. Prompt Claude to find the `wp_posts` INSERT row matching the target page by title or slug, and return the `post_content`, `post_title`, and `post_name`.
- 3 **WordPress MCP — `get_pages`** — Claude calls `get_pages` via the connected MCP to find the current page ID on the live site, matching by slug or title.
- 4 **WordPress MCP — `update_page`** — Claude calls `update_page` with the recovered `post_title` and `post_content` from the dump. No REST API, no curl, no credential setup.
- 5 **Verify in WordPress admin** — Confirm the page is live, check slug and publish status, and publish if restored as draft.

06 · TECH STACK

Claude AI

WordPress MCP

WP Engine Backups

SQL / database dump

`wp_posts` table

Full restore wipes live content

Large SQL files may need chunking for Claude

MCP must be pre-configured & connected

HONEST REFLECTION

The WordPress MCP made the write-back trivially easy — no auth setup, no curl, just a Claude tool call. The harder part was the SQL extraction. I'd write a short script to pull the target page row out of the dump automatically rather than feeding the whole file to Claude. The MCP approach is now my default for any WordPress content recovery; the days of manual REST API fiddling are behind me.

Read the full write-up — how this happened, why MCP changed everything, and what you need to replicate it.

[Read the blog post →](#)

To download more case studies, go to pranoti.thesciencetalk.com