Muhammad Asad · AI Implementation & Growth

Case Study · N° 07

DATA / PYTHON

Scraped a rival's full catalog, with receipts

Reverse-engineered a competitor's API to lift its entire 651-product catalog into a clean, audited dataset, every record carrying a SHA256 receipt of where it came from. The full dataset is downloadable below.

Mine · scraper, dataset, and analysis

DISCIPLINE	Data / Python
STACK	Python • httpx • Playwright (evidence only) • REST API reverse-engineering • SHA256 chain-of-custody • saturation + dedupe • data normalization

MY ROLE

Competitive-intelligence work behind the Cognilium legal-directory strategy. The scraper, the dataset, and the analysis are mine.

651	86%	51	10
PRODUCTS, FULL CATALOG	BOT TRAFFIC CAUGHT	API ENDPOINTS PROBED	SATURATION PASSES

THE PROBLEM

We needed the full, defensible competitive map of the legal-AI directory space, real data with a chain of custody, not guesses and not a scrape we had to take on faith.

WHAT I DID

- Reverse-engineered DreamLegal's API instead of scraping HTML: the site is a Next.js app with a public JSON API, so I read the endpoints directly. Probed 51, found the 5 that carried 100% of the data (product list, per-product score, reviews, taxonomy, blogs).
- Caught the API lying: every full pass returned a random ~65% sample of the catalog while always reporting a total of 651, so one scrape looked complete but quietly missed a third. I wrote a saturation scraper (paginate, dedupe by id, stop after consecutive zero-gain passes); 10 passes converged exactly to 651.
- Made every record auditable: ~1,700 polite requests at ~1 req / 0.5s, identifiable user-agent, robots respected, and an immutable raw layer where each response is stored verbatim with a SHA256 hash and full request/response metadata. The normalized layer regenerates from raw.
- Cross-checked 3 more platforms (Legaltech Hub, LawNext, r/legaltech) by HTML parsing under the same evidence discipline, 2,791 more vendor entities.

THE RESULT

A complete, auditable 651-product dataset, every record traceable to a hashed raw response, downloadable below. Building it also surfaced two things the raw numbers hid: an elaborate scoring 'moat' on ~0 real engagement, and an apparent market leader running ~86% bot traffic.

THE JUDGMENT CALL · WHAT THE AI COULDN'T DO

Two catches the raw data hid. First, the API was non-deterministic: it returned a random ~65% slice each pass while reporting a fixed total, so a normal scrape silently missed a third and still looked complete. I only caught it because the unique count drifted between runs; the fix was a saturation scraper that converged to the true 651. Second, the apparent market leader's traffic was ~86% synthetic. The data was right; the obvious read was wrong. Catching both was the whole value.

PROOF

Download: Full 651-product dataset + the methodology (below).

On request: Scraper toolchain + the per-platform analysis reports.

Muhammad Asad · AI Implementation & Growth Operator asad-portfolio-12.vercel.app · ma.asad412@gmail.com