

MCP SERVER

NO CODE

CLOUD HOSTED

# DatoCMS MCP for AI Agents

## Manage Headless Content and Digital Experience Platforms

DatoCMS gives your AI client direct access to your headless CMS data. You can talk to it naturally to discover content schemas via GraphQL queries, create new records, patch existing content blocks, and manage media assets—all without writing boilerplate code.

**A+** Quality Score 100/100

graphql

digital-experience

content-modeling

api-first

media-optimization

web-development



# The connectivity layer between AI and the world's software.



Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

# Your AI Connections Run Through Vinkius Cloud

The world's largest  
managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

*The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*

— Architecture principle

---

## Four Pillars of the Vinkius Runtime

### 01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

### 03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

### 02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

### 04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

**AES-256**

Encryption at rest

**Ed25519**

PKI vault signatures

**24h TTL**

Ephemeral session keys

**V8 Isolate**

Sandboxed execution

---

## One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

`https://edge.vinkius.com/{token}/mcp`

Claude



Cursor



VS Code



Windsurf



Grok



Gemini

---

## Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

### 01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

### 02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

### 03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

### 05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

### 04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

### 06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot is used outside Vinkius infrastructure, the server is quarantined instantly.

## Emergency Kill Switch

EU AI Act Art. 14(1)  
Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

#### 01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

#### 02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

#### 03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

## Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

**Control Plane**

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

**FinOps**

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

**Firewall & DLP**

PII redaction activity, sensitive data protection counters, and security event timeline.

**Agent Activity**

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

**Tool Health**

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

**Incident Log**

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

Get started at [cloud.vinkius.com](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# DatoCMS MCP

10 tools available

Cloud-hosted on Vinkius

This MCP lets you treat your entire digital experience platform like a natural language interface. Instead of jumping between the DatoCMS dashboard, running complex API calls in Postman, or remembering specific GraphQL query syntax, you just ask your agent what you need.

You can list every content model available, retrieve specific records by structure, and even update fields using simple commands. Need to delete a document node? It's handled. Want to check if an uploaded image is correctly linked through Imgix? Your agent handles the media oversight for you. Everything runs against your live DatoCMS environment, giving you full control over content creation, versioning, and structure definition. Because all of this is centralized on Vinkius, you connect once from any compatible client and immediately gain access to deep CMS management tools.

---

## Core Capabilities

### 01 — Querying Content Structure

Run complex GraphQL queries against your entire DatoCMS content graph to discover specific data arrays or schemas.

### 03 — Updating Existing Content

Modify existing document nodes or patch specific attribute blocks within a record without overwriting the whole entry.

### 05 — Managing Media Assets

List, retrieve details about, or validate uploaded media files to confirm their physical storage status.

### 02 — Creating New Records

Generate and provision entirely new CMS records, ensuring the payload strictly adheres to established JSON:API rules.

### 04 — Deleting Content Nodes

Irreversibly remove entire content records and associated data from the database.

### 06 — Auditing Content Models

Enumerate all explicitly defined content models and item types that structure your site's data.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/datocms](https://vinkius.com/mcp/datocms) — connect your AI agent in three steps.

- 01 Subscribe to this MCP on Vinkius.
- 02 Paste your DatoCMS Full Access API Token into the connection settings.
- 03 Use any MCP-compatible client (like Cursor or Claude) to issue a natural language request, and the agent executes the necessary content operations.

The bottom line is, you talk to your AI client, it talks to DatoCMS, and you get back structured data or confirmation of the change.

---

## Built For

This MCP targets developers and operational teams who spend too much time manually testing GraphQL queries, running repetitive content updates, or auditing media libraries across different environments. It's for anyone whose job involves interacting with a headless CMS structure.

### Front-end Developer

Testing complex data fetching scenarios by asking the agent to execute specific GraphQL queries against content models directly from their IDE.

### Content Editor

Performing routine maintenance, such as creating new blog posts or updating media assets, without having to leave the chat interface and switch between multiple CMS tabs.

### Digital Operations Engineer

Auditing content versioning across different environments and verifying record consistency by listing global models and checking API payloads.

## What Changes When You Connect

- 
- 01 Avoid writing repetitive data fetching code. Need content details? Ask the agent to run `get_single_record` instead of building a new query every time.

---

  - 02 Handle complex versioning tasks easily. Use `list_global_models` to audit what structures exist before you try to patch or create anything.

---

  - 03 Save time on media management. Instead of manually checking file paths, the agent can run `list_media_uploads` and verify asset status immediately.

---

  - 04 Perform bulk content changes safely. You don't need multiple endpoints; use `patch_cms_record` to modify specific fields across many records in one go.

---

  - 05 Improve developer workflow by eliminating manual querying. Run `execute_graphql_cda` directly through chat to test routing spaces without leaving your IDE.
- 

---

## Real-World Applications

### A new blog post needs structured data

A content editor wants a draft posted immediately. They prompt the agent, and it uses ``create_cms_record`` to provision the new item, ensuring all required fields like author ID and category are populated correctly based on schema validation.

### Troubleshooting media links

A front-end developer encounters a broken image. They instruct their agent to use ``get_media_upload`` on the component ID, which returns the exact structural match and verifies if the asset is correctly linked through `Imgix`.

### Content is outdated or needs a field change

The product team finds an old article missing key data. They ask their agent to use ``patch_cms_record`` to update the title and add a new metadata block across fifty existing records, all in one prompt.

### Discovering content structure limitations

A digital ops engineer needs to know what fields are available for a new site section. They ask the agent to run ``list_global_models`` and immediately get an exhaustive list of all possible item types and their definitions.

---

## Patterns to Avoid

---

### Treating content like static data

#### ✗ AVOID

Writing a script that just fetches the title of every post, ignoring which model or type it belongs to. This approach fails when your CMS structure changes.

#### ✓ INSTEAD

Always start by running ``list_global_models`` to understand the current schema boundaries. Then, use ``execute_graphql_cda`` to scope your queries correctly before fetching any content.

### Over-relying on manual API calls

#### ✗ AVOID

Having to write and test a separate CURL command in Postman every time you want to list records or check media status. This is slow and error-prone.

#### ✓ INSTEAD

Use the agent's natural language interface. For instance, ask it to 'list all recent media uploads,' which uses ``list_media_uploads`` and returns the data instantly.

### Deleting content without confirmation

#### ✗ AVOID

Running a deletion script that vaporizes multiple nodes without verifying if those nodes are still referenced by active pages or other critical systems.

#### ✓ INSTEAD

Before deleting, use ``list_cma_records`` to review the logs and confirm what data is currently pointing to the node. Only then should you authorize the irreversible action using ``wipe_cms_record``.

## The Right Fit

Use this MCP if your primary pain point is interacting with a complex, structured headless CMS (like DatoCMS) from an AI agent or IDE. You need to programmatically read schemas ( `list_global_models` ), manipulate records ( `create_cms_record` , `patch_cms_record` ), and audit media assets without writing dedicated API wrappers in your code base. Don't use this if you just need simple data storage, like a basic key-value database; those require simpler integration types. Also, don't use it if you only ever read public content—you need the full write/mutate capabilities to make it worth connecting.

---

## DatoCMS MCP for AI Agents: Managing Headless Content Structures

Today, managing a headless CMS means constant context switching. You open your dashboard to check the content models, switch tabs to run a GraphQL query in an IDE, copy the result into a spreadsheet, and then write another script just to patch a single field on one of those records. It's tedious, slow, and easy to lose track of which version you're actually modifying.

With this MCP, you talk to your agent instead. You tell it, 'Update the Hero section title for Product X.' The agent handles listing models, validating the record type, executing a `patch_cms_record` , and confirming success—all in one turn. What you get is immediate, reliable content mutation without leaving your chat window.

---

## DatoCMS MCP for AI Agents: Auditing Digital Experience Platforms

Auditing a live DXP involves deep dives into asset management. You have to manually check the Imgix proxy mappings, list every single media upload, and cross-reference those IDs with the content records themselves just to ensure nothing is orphaned or broken.

Now, you simply ask your agent to 'Show me all unused media assets.' It uses `list_media_uploads` and identifies exactly what needs clearing. You gain total visibility into the asset pipeline, making cleanup instant and verifiable.

---

# DatoCMS: 10 Tools for Headless CMS Content Modeling

Use these tools to perform everything from creating new content items to running complex GraphQL queries against your entire DatoCMS platform structure.

#	TOOL	DESCRIPTION
01	<code>create_cms_record</code>	Generates a new, fully structured JSON payload to provision fresh content items into the CMS.
02	<code>wipe_cms_record</code>	Deletes an explicit content record and all associated live data rows from the database permanently.
03	<code>wipe_media_upload</code>	Runs a validation check that automatically removes specific, unused disk assets.
04	<code>get_single_record</code>	Extracts the full set of properties for one specific content node based on its unique identifier.
05	<code>get_media_upload</code>	Retrieves detailed structural information about a specific uploaded media file block.
06	<code>execute_graphql_cda</code>	Runs targeted GraphQL queries to identify bounded routing spaces within the DatoCMS content graph.
07	<code>list_global_models</code>	Provides a list of all explicitly registered and attached structured rules that define item types in the CMS.
08	<code>list_cma_records</code>	Retrieves explicit cloud logging data, showing entire JSON:API arrays for audit purposes.
09	<code>list_media_uploads</code>	Inspects all internal media arrays to check on specific image assets and their metadata.
10	<code>patch_cms_record</code>	Safely updates a record by modifying or substituting specific item parameters within an existing content block.

---

## See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

**U** I need a list of all content model types available for this site.



Sure thing. I found 6 content models:

- **Blog Post:** Defines article structure (Title, Body, Author).
- **Product Page:** Requires SKU, main image, and description.
- **Author Profile:** Needs name, bio, and social links.
- **Category:** Simple listing model.
- **Team Member:** Captures photos and roles.
- **Page:** The generic container model.

Which one should we check records for?

**U** Can you update the main hero image on the Product Page template?



I've updated the featured product banner. I used `patch cms record` to replace the old asset with the new high-res photo, and confirmed that all associated meta tags are correctly mapped. The record ID for this was 456.

**Status:**  Updated Successfully

**Details:** New Image URL: `https://cdn.datocms.com/new-hero.jpg`

## U List all the media uploads that haven't been used in three months.



I ran a check on your asset library and found 12 assets flagged as unused based on their last access date.

ASSET NAME	ID	LAST USED DATE
old-logo-v1.png	901	2023-01-15
team-photo-backup.webp	882	2023-04-20

Do you want me to run `wipe_media_upload` on these?

## Frequently Asked Questions

### 01 How does DatoCMS MCP help me update content without complex GraphQL queries?

You simply tell your agent what you want changed, like 'Change the author of Blog Post 123.' The MCP handles all the underlying complexity, running the necessary patch commands and ensuring data integrity for you.

### 02 Can I use DatoCMS MCP to see what content models my site supports?

Yes. You can ask the agent to list all global models. This gives you an immediate inventory of every defined content type, which is essential when onboarding a new team member or architecting a new feature.

### 03 What if I need to delete old records in DatoCMS?

The MCP allows for irreversible deletion. You can specify the record you want gone, and the agent will execute the node vaporization process, clearing out unused data safely.

### 04 Is this better than just using the DatoCMS web interface?

It is faster and more repeatable. The MCP lets you run complex audits or batch updates in seconds via chat, whereas doing that manually in the dashboard takes hours of clicking through different screens.

### 05 Does DatoCMS MCP handle media asset cleanup for me?







Yes, it provides tools to inspect and manage your assets. You can list all uploads and even run automated checks to remove unused or obsolete files from the system.

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://cloud.vinkius.com), then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"datocms": { "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	Extensions → Add MCP Server → Paste endpoint URL

## ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

-  **Ask ChatGPT** 
-  **Ask Claude** 
-  **Ask Perplexity** 
-  **Ask Gemini** 
-  **Ask Grok** 

READY TO CONNECT

# DatoCMS is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

[Start at cloud.vinkius.com](https://cloud.vinkius.com) →

[vinkius.com](https://vinkius.com) · [support@vinkius.com](mailto:support@vinkius.com)

### INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by DatoCMS. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

### DOCUMENT INFORMATION

Generated	June 2026
MCP Server	DatoCMS MCP
Server ID	019d7582-9997-7110-a6c5-ef639cc98e85
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit [vinkius.com/mcp/datocms](https://vinkius.com/mcp/datocms).