MCP SERVER | NO CODE | CLOUD HOSTED

Drupal MCP for AI Agents

Orchestrate headless Drupal content and JSON:API workflows

Drupal MCP gives your AI clients full control over headless content management, letting you interact with complex CMS structures and JSON:API endpoints naturally. List nodes, manage taxonomy, handle files, and audit users without touching the admin interface.

A+ Quality Score 100/100

json-api | node-management | taxonomy | entity-framework | web-development | content-orchestration

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/drupal

SCAN TO VIEW

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.

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.

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.

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	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	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.

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.

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.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken 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.

Agent Activity

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

FinOps

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

Tool Health

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

Firewall & DLP

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

Incident Log

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

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Drupal MCP

10 tools available
Cloud-hosted on Vinkius

This connection lets you treat your Drupal site like any other data source, managing all its structured content through simple conversation. You can provision new articles or update existing ones by simply telling your agent what needs changing, bypassing traditional forms entirely. Need to see which files are attached to a specific article? Just ask. The system maps out the entire internal structure—nodes, categories, and media assets—and makes it available instantly. Because Vinkius hosts this MCP within its catalog, you connect once from your preferred AI client and get immediate access to Drupal's full content lifecycle management tools.

It’s about getting hands-on control over the backend data layer so developers can test APIs or marketers can update categories without needing administrator credentials.

Core Capabilities

01 — Provisioning Content Nodes

The agent writes new, structured content into Drupal entities using a JSON payload.

02 — Deleting Articles and Pages

You can irreversibly remove live content nodes directly from the system.

03 — Discovering File Assets

The MCP retrieves detailed information about managed files, including their location and metadata.

04 — Reading Specific Content Records

It fetches the complete details for any specific content node by its unique ID.

05 — Managing Categories and Terms

The agent extracts or lists structured categories (taxonomy) that define how your content is grouped.

06 — Listing All Content Structures

It identifies all available types of content nodes within the Drupal model.

07 — Auditing Site Users

You can retrieve a list of registered user accounts, showing roles and access levels.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/drupal** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Drupal Base URL along with the necessary authentication header. Make sure the JSON:API module is active on your site.

02 Connect your AI client (like Cursor or Claude) to Vinkius, giving it access to this content management tool.

03 Start by asking your agent to perform an action, like listing all available categories or creating a new article draft. The system handles the rest.

The bottom line is that you interact with Drupal's complex backend data using natural language commands instead of filling out forms or writing boilerplate API calls.

Built For

This MCP targets technical roles and content creators who need programmatic control over a robust CMS. It's for the developer debugging an endpoint in their IDE, or the editor needing to update categories faster than using the admin UI.

Digital Marketing Manager

Uses the MCP to categorize and update content assets across the site without ever logging into Drupal's backend.

Content Editor

Creates new articles or modifies existing drafts, managing both the body text and associated category terms through conversation.

Full-Stack Developer

Tests JSON:API endpoints and verifies entity schemas directly from their IDE by querying specific content nodes or listing available fields.

What Changes When You Connect

01 Update content structure without the admin UI. Use `create_cms_node` or `patch_cms_node` to write new articles or modify existing drafts directly via conversation.

02 Audit site users instantly. The `list_drupal_users` tool provides a clean roster of all editors and admins, letting you manage who has access without manual database queries.

03 Manage media assets easily. Instead of navigating file directories, use `get_file_metadata` or `list_managed_files` to pull raw CDN URIs for any attachment.

04 Control content structure at scale. You can list all available content types using `list_content_nodes`, giving you a complete map of your site's data architecture.

05 Maintain organizational integrity. The MCP lets you enumerate categories using `list_term_vocabularies` and pull details for specific terms with `get_taxonomy_term`.

Real-World Applications

Updating a product category across multiple articles

A marketer needs to change the primary taxonomy term for 20 old blog posts. Instead of manually opening each post and updating the tag, they ask their agent to update the category mapping across all relevant nodes using `get_taxonomy_term`.

Debugging a broken content display endpoint

A developer finds an API endpoint failing. They use the MCP to pull a specific node's data (`get_single_node`) and cross-reference it with file metadata using `get_file_metadata` to pinpoint exactly which attachment is causing the failure.

Generating initial draft content for launch

A team leader needs 10 new articles written quickly. They prompt their agent to create these drafts (`create_cms_node`) using a provided JSON payload, populating the required fields and nodes instantly.

Reviewing who can edit content

An ops team member needs an immediate audit of site access. They ask the agent to list all users (`list_drupal_users`) so they can verify which accounts have editor rights before a major deployment.

Patterns to Avoid

Treating Drupal like a simple database

✗ AVOID

Asking the agent to just 'get all content.' This fails because Drupal data is structured and categorized, not just in flat rows. You need context.

✓ INSTEAD

Always start by asking for a list of content types using `list_content_nodes` or listing available categories with `list_term_vocabularies`. This defines the scope you're working within.

Trying to update data without knowing the ID

✗ AVOID

Telling the agent, 'Update the article about marketing.' The system can't find it because content nodes need a unique identifier to patch safely.

✓ INSTEAD

Use `get_single_node` first with the specific UUID or node ID. Once you have that reference, use `patch_cms_node` to apply targeted changes.

Ignoring file dependencies

✗ AVOID

Creating a new article and forgetting to mention an image needs uploading. The resulting content will be broken because the link is missing.

✓ INSTEAD

Run `list_managed_files` first to confirm all necessary media assets exist, then use those references when calling `create_cms_node`.

The Right Fit

Use this MCP if your primary need involves managing the structured data layer of a headless Drupal site. If you are constantly writing content nodes, updating categories, or auditing user roles programmatically, this is for you. Don't use it if you only need to read simple blog posts; in that case, standard public API calls might suffice. You should *not* use this MCP if your content isn't managed by Drupal, regardless of how robust the CMS is. Furthermore, if you just need basic search functionality without write access, you don't need these tools; you'd look for a read-only endpoint.

Managing Drupal Content Nodes with AI Agents

Manually managing content in Drupal means clicking through node lists, opening separate category forms, and then navigating to media libraries. Copying data from one place to another, or updating a single field across dozens of articles, is tedious and prone to human error.

With this MCP, the same task becomes conversational. You tell your agent you need to create an 'article' about X with Y categories and Z associated images. It handles the entire workflow—provisioning the node, linking the taxonomy, and attaching the media—and gives you a fresh, live document entity.

Auditing Drupal User Roles via AI Agents

Traditionally, checking user permissions requires navigating complex role management screens or querying internal databases with specific SQL commands. It’s a manual audit that takes time and specialized knowledge.

Now, you simply ask the agent to list all registered users. You get an immediate, structured report detailing every account, their assigned roles, and access level—all without leaving your chat interface.

10 Tools for Drupal Content Orchestration and Taxonomy

Use these tools to create articles, delete nodes, list users, and manage all structured data points within your Drupal instance.

#	TOOL	DESCRIPTION
01	`create_cms_node`	Writes a new piece of structured content into the Drupal system.
02	`wipe_cms_node`	Permanently deletes an existing content node, removing it from the live site data.
03	`get_file_metadata`	Retrieves structural details about a specific file stored in Drupal's system.
04	`get_single_node`	Pulls the full content and metadata for one particular article using its unique ID.
05	`get_taxonomy_term`	Extracts all properties associated with a specific category or term mapping.
06	`list_managed_files`	Inspects and lists all managed files available on the site, useful for checking media assets.
07	`list_content_nodes`	Identifies and returns a list of all types of content nodes that exist in the Drupal model.
08	`list_term_vocabularies`	Lists all available sets of structured rules used for categorizing content (taxonomies).
09	`list_drupal_users`	Retrieves a list of every registered user account, along with their roles and access levels.
10	`patch_cms_node`	Safely updates specific segments or values within an existing content node draft.

See It in Action

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

U What are the most recent articles on 'Technology' that need an update?

Recent Content Nodes in 'Technology'

UUID	TITLE	STATUS	LAST MODIFIED
abc-123	Drupal MCP Guide v2.0	Published	2 hours ago
xyz-456	Next Gen Headless CMS Trends	Draft	Today, 9:15 AM

Do you want me to pull the full content for any of these?

U I need to create a new article about best practices. Can you provision it?

Node Creation Request:

- **Title:** Best Practices for Headless Drupal
- **Body:** [Draft text provided by user]
- **Categories:** ['Best Practice', 'Developer']

✅ Success! The new node was created with UUID: `d3f-90b`. You can view it in your frontend now.

U List all the files attached to the 'API Basics' guide.

File Metadata for API Basics (UUID: 456)

Image Asset: `api_diagram.png` *(CDN URI:* `/cdn/media/12345.png` *)* - Status: Available*
Logo: `drupal_logo.svg` *(CDN URI:* `/cdn/media/67890.svg` *)* - Status: Deprecated*

You have two assets attached. Which one do you want me to fetch the full dimensions for?

Frequently Asked Questions

01 **How do I manage content nodes in Drupal using the Drupal MCP for AI Agents?**

You can treat content node management like a natural conversation. Instead of going through forms, you ask your agent to create drafts, update existing articles, or modify specific fields directly by referencing the data.

02 **Can I use Drupal MCP for AI Agents to audit all user accounts?**

Yes. You can request a complete list of registered users and their associated roles. This allows you to verify who has editor access across your entire site structure without accessing the admin panel.

03 **What if I want to delete an old article? Can Drupal MCP for AI Agents do that?**

The MCP can irreversibly remove content nodes. If you provide the correct unique ID, the agent will vaporize the node from your live site data.

04 **How does this help with media and file attachments in Drupal?**

You don't have to guess where files are located. The MCP lets you list all managed files and retrieve detailed metadata, giving you the exact CDN URIs for every image or asset.

05 **Does Drupal MCP for AI Agents handle content categorization?**

Absolutely. You can enumerate all available categories (taxonomies) and get specific details on how content is grouped using this connector, making content organization programmatic.

Go Live in 60 Seconds

Get your connection token from **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
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"drupal": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	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

Drupal 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 →

vinkius.com · 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 Drupal. 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	Drupal MCP
Server ID	`019d7589-ff77-70f5-a357-9766b16965f0`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

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/drupal.

© 2026 Vinkius · vinkius.com · support@vinkius.com