MCP SERVER | NO CODE | CLOUD HOSTED

ContentStack Management MCP for AI Agents

Automate Content Publishing and Schema Updates in Headless CMS

ContentStack Management gives your AI agent full read and write control over your headless CMS. Stop clicking through web dashboards; use conversational commands to create new content, update existing records, verify schemas, and push finalized material from staging directly to global production.

A+ Quality Score 100/100

content-api | cms-automation | workflow-management | schema-updates | headless-architecture

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/contentstack-management

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.

ContentStack (Management) MCP

10 tools available
Cloud-hosted on Vinkius

Managing a large digital site usually means juggling multiple dashboards—one for drafting, one for metadata, and another for deployment status. This MCP changes that process entirely. It lets your AI agent talk directly to the ContentStack core, giving it full read-write access over your content structure and environments. Instead of manually logging into an admin panel to build a new article or fix outdated data points, you just tell your agent what needs doing in plain text. Your agent can map complex variables, create brand new entries for any defined type, fetch huge amounts of metadata across your entire stack, and most critically, it handles the full publishing cycle. You'll manage content logistics from scratch to final deployment without ever touching a clunky web UI. When you connect this through Vinkius, you get that power centralized in one place.

Core Capabilities

01 — Discover Content Structures

List all existing content types and retrieve detailed schema information to understand how your data is structured.

02 — Manage Published Media Assets

Browse and list all media assets stored in the stack, helping you track images and files used across entries.

03 — Read and Write Content Records

Get full details for any specific entry or list multiple draft records belonging to a content type.

04 — Create New Digital Entries

Generate entirely new entries by passing required parameters like title, body text, and related tags.

05 — Update Existing Content Fields

Modify specific fields or update an entire entry record that already exists in the system.

06 — Control Deployment Environments

List all available publishing environments and push a finished content piece from staging directly into production.

One Click on Vinkius — From Prompt to Execution

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

01 Connect your AI client to this MCP using an overarching API Key alongside a dedicated Management Token.

02 The agent uses the connection to query the ContentStack core, pulling necessary schema and content data into its working memory.

03 You issue a conversational command—like 'Publish entry X to production' or 'Build new blog post about Y'—and your agent executes the required write operation.

The bottom line is that you talk to your AI client, and it talks to ContentStack for you, executing complex CMS actions without any manual dashboard interaction.

Built For

ContentStack Management targets technical roles who deal with content at scale. It's built for the developer frustrated by slow deployment pipelines, the marketer needing bulk updates across hundreds of pages, and the architect building custom data ingestion workflows.

Content Marketing Manager

Needs to generate new blog posts or update product descriptions in bulk based on campaign requirements without touching the CMS UI.

Technical Writer / Editor

Must audit and correct metadata, ensuring every entry has the right tags or belongs to the correct taxonomy before publication.

Full-Stack Developer

Uses it programmatically to map nested data structures, ingest seed content during staging, or automate environment verification steps.

What Changes When You Connect

01 Skip the dashboard overhead. Instead of manually navigating menus to update an item, you can simply tell your agent to use `update_entry` with a specific ID and new data fields.

02 Gain full visibility into your content structure using `list_content_types`. You instantly know what schemas exist without checking documentation or guessing names.

03 Never worry about deployment lag again. With the ability to call `publish_entry`, you can move finalized drafts from staging directly to global production with a single command.

04 Scale data ingestion massively. If you need to build 100 product pages, your agent handles the bulk process of calling `create_entry` repeatedly, saving hours of manual work.

05 Centralize auditing and status checks. Use `get_stack_info` or list assets to keep a clear record of what's live, what's draft, and where it lives.

Real-World Applications

Migrating old content into new schemas

A developer needs to take 500 legacy articles written in an old format. They ask their agent to first use `list_content_types` to confirm the target schema, and then loop through all existing IDs using `list_entries` before calling `create_entry` for each one with mapped data.

Fixing a broken live page

The marketing team notices a product listing has outdated pricing. They tell their agent to use `get_entry_details` on the specific UID, correct the price field in the response, and then call `update_entry` before asking for immediate publishing.

Running pre-launch environment checks

The project manager asks their agent to use `list_environments` to confirm all necessary stages exist. Then they instruct the agent to run a test publish on several key entries, validating that `publish_entry` works across multiple nodes.

Generating seed content for testing

A developer starts a new feature and needs mock data. They ask their agent to use `create_entry` repeatedly, specifying titles and body text, generating dozens of dummy articles instantly for testing the front end.

Patterns to Avoid

Assuming content is ready to publish

✗ AVOID

A user manually drafts a post in their editor and then tells the agent 'Publish it.' This fails because the entry might be sitting in a draft status, not fully validated.

✓ INSTEAD

Always verify the state first. Tell your agent to run `get_entry_details` to confirm all required fields are populated, then use `publish_entry` only when you're sure the data is ready for the target environment.

Updating content without knowing its type

✗ AVOID

A user tries to update an entry using general text prompts, but fails because they didn't first confirm if the article belongs to `blog_post` or a custom schema.

✓ INSTEAD

Start by running `list_content_types`. This shows you all available schemas. Then, use `get_content_type_details` on the correct type before attempting to find and update any entry.

Overwriting everything with a single command

✗ AVOID

A user asks the agent to 'Fix all data.' This is too vague and could corrupt important metadata across hundreds of entries.

✓ INSTEAD

Be specific. Use `list_entries` to identify the exact group of UIDs you need to fix, then use `update_entry` only on those confirmed IDs for targeted fixes.

The Right Fit

Use this MCP if your workflow demands programmatic control over content lifecycle: generating data, modifying schemas, and managing environments. You'll want it when the task is 'how do I get X state to Y state?' If you only need to read simple reports or view assets without making changes, a basic retrieval-only tool might suffice. Don't use this if your goal is just to draft text; that requires a language model. Use `create_entry` when you know the structure and want new content. But if you are unsure what structures exist in your CMS at all, start with `list_content_types` first.

ContentStack Management MCP for AI Agents: Schema Modeling Pain Points

Right now, updating complex content means jumping through hoops. You have to switch between the CMS UI and a separate metadata editor just to verify that every single field—the taxonomy, the author ID, the featured image—is correct before you can hit 'Save.' It's tedious clicking, copy-pasting IDs from one panel into another, and hoping you didn't miss a required tag.

With this MCP connection, your agent handles the whole flow. You simply ask it to update an entry, providing all the parameters in a single prompt. The agent uses `update_entry` behind the scenes, ensuring every part of that complex record is correctly addressed and saved—you just get the final confirmation.

ContentStack Management MCP for AI Agents: Publishing Workflow Control

Manual publishing pipelines are risky. You have to remember if you pushed it from 'staging' or 'preview,' and whether the global production layer is actually accepting new content. It’s a manual checklist that nobody wants to repeat, especially after hours.

Now, your agent manages the entire deployment chain. After drafting a piece, you prompt the system to publish it, and the agent executes `publish_entry`, moving it from wherever it sits directly to the live environment without any human intervention or guesswork.

ContentStack (Management) 10 Content Schema Tools for Developers

These tools give your agent the ability to read, write, list, and publish content entries across all defined schemas in your ContentStack environment.

#	TOOL	DESCRIPTION
01	create_entry	Generates a brand-new entry and populates it with initial content parameters.
02	get_content_type_details	Retrieves the complete schema definition for any specific content type in your stack.
03	get_entry_details	Pulls all current data and metadata fields for a single, specified entry ID.
04	get_stack_info	Fetches general status information about the entire ContentStack setup, useful for auditing or monitoring.
05	list_assets	Provides a list of all media files and assets currently stored within your stack's library.
06	list_content_types	Lists every available content model (schema) that the system recognizes.
07	list_entries	Retrieves a list of all existing entries for a specified content type, often used for batch processing checks.
08	list_environments	Shows every defined publishing stage (like 'staging' or 'development') available in your ContentStack setup.
09	publish_entry	Forces an entry to move from its current environment and publish it to a specified target environment.
10	update_entry	Modifies the content or metadata of an existing entry that you already know the ID for.

See It in Action

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

U I need to write three new blog posts about modern development trends. Can you generate them using the 'blog_post' type?

Generating 3 draft entries for `blog_post` ...

✅ Success! Three drafts created:
- **Title:** The Future of LLMs in Code (UID: bltpxxxx101)
- **Title:** Advanced Schema Design Patterns (UID: bltpxxxx102)
- **Title:** Next-Gen Web Frameworks (UID: bltpxxxx103)

All three are available as drafts. Ready to publish them live?

U I just finished revising the product page for Acme Corp. Can you update entry 'bltpxxxx205' with this new body text and set the featured image?

Updating entry `bltpxxxx205` ...

➡️ **Status:** Success.
- **Fields Updated:** Body content (new text length: 4,120 characters). Featured Image (Asset ID: 9876).
- **Metadata Check:** Taxonomy updated to 'Product', status remains Draft. The entry is ready for review.

U I want to make sure the new marketing page gets live. Please move entry `bltpxxxx300` from staging to production.

Initiating deployment protocol for UID `bltpxxxx300` ...

🔥 **Deployment Status:** Complete.

- **Source Environment:** Staging
- **Target Environment:** Production (Global)
- **Timestamp:** 2026-06-17T14:30:00Z

The content is now live and accessible globally.

Frequently Asked Questions

01 **How do I use ContentStack Management MCP for AI Agents to create new blog posts?**

You tell your agent the title, body text, and target content type. The agent uses the underlying tools to generate a draft entry immediately, giving you a unique ID that you can then review or publish later.

02 **Can I use ContentStack Management MCP for AI Agents to fix outdated product descriptions?**

Absolutely. You give the agent the specific content type and the IDs of the entries needing fixes. The agent updates the fields, ensuring consistency across all your products in one go.

03 **What is the easiest way to push a finalized article from draft status to live?**

You use ContentStack Management MCP for AI Agents to call the publishing tool. This bypasses manual clicks and moves the entry directly from its current testing environment straight into global production.

04 **Does this MCP help me see what schemas I already have in my CMS?**

Yes, you can use ContentStack Management MCP for AI Agents to list every content type available. This is essential for knowing which structured data models exist before you write new content.

05 **What if I need to bulk update hundreds of entries at once?**

The agent handles batch operations. You can use the listing tools and then loop through them, telling the agent to run updates on many IDs simultaneously for massive data cleanup or restructuring.

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 `"contentstack-management": { "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

ContentStack (Management) 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 ContentStack (Management). 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	ContentStack (Management) MCP
Server ID	`019d757b-2dd7-72b2-9a04-a0d353f4fc01`
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/contentstack-management.