MCP SERVER | NO CODE | CLOUD HOSTED

Contentstack MCP for AI Agents

Query live content entries and schemas from a headless CMS

Contentstack gives your AI agent instant read access to published digital content and schemas from a headless CMS. Your agent can search for live articles by title, pull specific asset URLs (like images or PDFs), and audit the underlying data structure of any content type in real time.

A+ Quality Score 100/100

api-delivery | content-retrieval | omnichannel | schema-management | content-delivery-network

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

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 MCP

9 tools available
Cloud-hosted on Vinkius

This MCP connects your AI client directly to Contentstack's Delivery API, letting your agent read your published digital assets securely. Instead of relying on outdated documentation, your agent can pull live information from your entire stack—whether that's a specific blog post entry or the schema definition for a new content type. You can ask it to find the exact URL for an image used in a campaign, or search through thousands of entries looking only for those tagged 'Q3 2024 Marketing'. It's like giving your agent real-time eyes into your entire publishing environment. If you use Vinkius, this MCP adds Contentstack functionality alongside hundreds of others, keeping all your critical data sources in one place.

It handles everything from high-level metadata checks to detailed structural inspections, making it useful for developers who need live payload structures or content managers needing quick audits of 'staging' versus 'production' copies.

Core Capabilities

01 — Query published articles by title or filter

Your agent reads specific blog posts or entries based on provided titles or search criteria.

02 — Retrieve media asset URLs and details

The MCP finds the direct, usable links for images, PDFs, or other files stored in your content library.

03 — Inspect content type schemas

Your agent reads the blueprint (schema) of any content type so you know exactly what data fields are available to use.

04 — List all published assets and types

You can ask the MCP to list every asset or every defined content structure within your stack.

05 — Search through massive entry sets

The agent performs deep searches across multiple entries using complex JSON queries for highly specific data points.

One Click on Vinkius — From Prompt to Execution

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

01 Link your Contentstack workspace to the MCP using a read-only Delivery Token and your Stack API Key.

02 Your agent sends natural language requests, specifying what content or assets it needs (e.g., 'Find the hero banner text').

03 The MCP executes the necessary query against Contentstack, returning validated data payloads like article bodies or asset URLs to your client.

The bottom line is, you talk naturally to your AI agent and get structured data pulled directly from your live publishing environment.

Built For

Developers who need live payload structures without leaving their IDE. Content Managers who spend hours auditing content across staging and production environments. Marketing Integrators needing to locate public URLs for cross-platform campaign assets quickly.

Web Developer

Uses the MCP to check live data structures instantly, verifying if frontend rendering code will break before deployment.

Content Manager

Directly queries content variants and entries across 'production' or 'staging' environments to confirm which copy is currently available for use.

Marketing Integrator

Asks the MCP to locate public URLs for campaign assets, enabling efficient distribution of visual materials across different channels.

What Changes When You Connect

01 Need to verify the structure of a new 'Hero Banner' model? Use `get_content_type_details` to read the schema blueprint before writing any front-end code.

02 Instead of manually checking multiple tabs, ask for all content variants in a specific environment using `list_entries`. It pulls the data instantly into your chat context.

03 Find that perfect image URL across thousands of assets. The MCP allows you to `get_asset_details` and pull direct CDN links without guessing file paths.

04 Don't waste time re-fetching everything. Use `sync_content` to tell your agent only what has changed since the last check, making queries faster.

05 Need a specific article? The MCP lets you search using `search_entries`, running complex JSON filters that go way beyond simple title searches.

Real-World Applications

Debugging frontend rendering

A developer needs to know if the 'Author Bio' field exists and what its data type is. They use `get_content_type_details` so they can verify payload structures without leaving their IDE.

Pre-launch content audit

A Content Manager needs to check if all 'Product Page' entries have a featured image and a body text. They use `search_entries` with filters, getting a clean list of missing data points.

Finding campaign assets for social media

A marketing team member needs the direct URL for the 'Spring Sale' banner image. They ask to `get_asset_details`, and the agent provides the exact, usable CDN link immediately.

Migrating content data

An integrator needs a baseline of all available structured content types before building an automated pipeline. They use `list_content_types` to map out every field they need to handle.

Patterns to Avoid

Assuming the data is structured

✗ AVOID

Writing code that expects a 'slug' field, but failing when Contentstack changes it to an internal ID. This causes runtime errors.

✓ INSTEAD

Always run `get_content_type_details` first. This verifies the schema and tells you the exact names (like 'external_id') your agent needs to use for reliable code.

Listing assets without filtering

✗ AVOID

Calling a list function that returns 5,000 generic files, forcing your client to sift through thousands of irrelevant images.

✓ INSTEAD

Use `list_assets` and follow up with `get_asset_details` specifying the asset ID. This narrows the focus immediately.

Relying on outdated content

✗ AVOID

Showing a client an old version of a blog post because they didn't check for updates.

✓ INSTEAD

When querying, always use `sync_content` first. This ensures your agent is working with the most recently published data in the target environment.

The Right Fit

Use this MCP if you need your AI client to interact with live, structured content data from a headless CMS—specifically when retrieving assets or validating schemas matters. Don't use it if you just need to chat about marketing strategy; that's for general LLMs. If your primary goal is only generating text based on prompts, you might be fine without this. But if the task requires reading *what* data exists (e.g., 'What fields are available?') or retrieving a specific asset URL (`get_asset_details`), then connecting Contentstack via Vinkius is essential.

Contentstack: Schema Auditing for Headless CMS Development

Currently, checking the available fields in your content model means jumping between the CMS dashboard and your IDE. You manually click through different content types to confirm if you can access a user's phone number or just their name.

With this MCP, you simply ask your agent: 'What schema does the Client Profile content type use?' The agent runs `get_content_type_details` and gives you the full, structured breakdown right in the chat window. You get immediate technical validation without leaving your development environment.

Contentstack: Retrieving Live Published Entries for Marketing

Before this MCP, pulling a set of campaign assets required someone to manually browse the media library and copy-paste dozens of URLs into a spreadsheet. It was slow and prone to broken links.

Now, you can ask your agent to find all entries matching 'Fall Campaign' that were published in 'staging'. The agent uses `search_entries` and provides clean data payloads, including direct asset references, ready for consumption.

Contentstack: 9 Tools for Content Retrieval & Schema Management

These tools let your agent inspect, list, search, and retrieve all types of digital assets and data structures within your Contentstack stack.

#	TOOL	DESCRIPTION
01	`get_asset_details`	Retrieves specific details about a single media asset by its ID.
02	`get_content_type_details`	Reads the structural schema definition for any given content type.
03	`get_entry`	Fetches all detailed, published content for a specific entry ID.
04	`get_stack_summary`	Retrieves high-level metadata about the entire Contentstack environment.
05	`list_assets`	Lists all available published media assets in your library.
06	`list_content_types`	Provides a list of every content structure defined within the stack.
07	`list_entries`	Lists all published entries belonging to one specific content type.
08	`search_entries`	Searches for and retrieves multiple entries using a complex, customizable JSON query.
09	`sync_content`	Checks for and pulls only the changes that have occurred in the content since your last successful sync.

See It in Action

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

U Pull the full text and featured image URL for the blog post titled 'Future Trends in AI' from the production site.

Content Found:
- **Title:** Future Trends in AI
- **Author:** Jane Smith
- **Published:** 2024-10-01
- **Summary:** The main text emphasizes LLM advancements mapped over the next five years.
- **Featured Asset URL:** `https://cdn.contentstack.com/assets/banner-ai-v3.jpg`

U What are the required fields for a 'Team Member' profile entry?

Content Type Schema: Team Member
- **heading:** Short text (Mandatory)
- **bio_text:** Long text (Optional)
- **avatar:** File reference (Mandatory)
- **department:** Text field (Optional)

U Show me the three most recently uploaded assets.

Here are the latest items from your library:

1. `Product-Logo-v2.svg` (Type: SVG, Size: 15KB)
2. `Banner-Spring-2024.png` (Type: PNG, Size: 890KB)
3. `AnnualReport_Q2.pdf` (Type: PDF, Size: 4MB)

Frequently Asked Questions

01 **How do I use the Contentstack MCP to find content schemas?**

You simply ask your agent what structure a specific content type needs. The MCP reads the underlying schema definition, so you know exactly which fields—like 'headline' or 'main_body'—you can expect when retrieving data.

02 **Can Contentstack help me find asset URLs for my website?**

Yes. You ask your agent to list assets, and it retrieves the direct CDN links for images, PDFs, or any other file stored in your content library so you can use them immediately.

03 **What if I need content from a specific environment like 'staging'?**

You guide your agent to target the correct environment name. This ensures that when you pull entries, you get the version of the content meant for review or testing, not what's live.

04 **Is Contentstack MCP good for large-scale data migration?**

Yes. It offers tools like `sync_content` and `list_entries`, letting you check only the delta of changes since your last sync, which makes migrating or auditing massive amounts of content much faster.

05 **Does Contentstack MCP help me search for specific articles?**

Absolutely. You don't have to rely on titles; you can use the agent to run complex JSON queries against entries, allowing you to filter results by author ID, date range, or custom tags.

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": { "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 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. 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 MCP
Server ID	`019d757b-14de-70f8-a4f8-84be65a757e3`
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.