MCP SERVER | NO CODE | CLOUD HOSTED

Storyblok MCP

Manage content structure and assets via conversation.

Storyblok MCP lets your AI client manage content structure and assets directly within Storyblok. It gives you programmatic control over headless CMS environments. You can list entire content spaces, analyze available component blueprints, draft new stories, update existing articles, and find all media files without ever touching the visual editor.

A+ Quality Score 100/100

content-modeling | visual-editor | api-driven-content | digital-experience | asset-management | structured-data

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

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.

Storyblok MCP

9 tools available
Cloud-hosted on Vinkius

Managing complex digital content used to mean clicking through endless dashboards or copy-pasting structural IDs from one tab to another. This MCP changes that. It connects your AI agent directly into Storyblok's backend, treating your entire content repository like a database you can talk to. You instruct your agent what you need—say, 'Give me all the components used in our marketing space and draft a new story using those elements.' The system finds the necessary blueprints, validates them against existing data, and then builds or updates the content for you. It’s about treating content not as visual blocks, but as structured data. If you're building out a complex site structure and need an AI assistant that understands component hierarchies and asset libraries, Vinkius has this MCP ready to go.

Core Capabilities

01 — Discover Content Spaces

Lists all accessible Storyblok environments so your agent knows where to start working.

02 — Inventory Stories and Assets

Retrieves lists of existing articles or scans the entire media library for multimedia files.

03 — Analyze Component Blueprints

Lists all defined content components, helping you understand what structural elements are available to build with.

04 — Build and Modify Content

Creates new articles or updates fields in existing stories using structured data inputs.

05 — Audit Space Users

Checks who has editing access to a specific content space, helping manage internal permissions.

One Click on Vinkius — From Prompt to Execution

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

01 Establish the Storyblok MCP module on your preferred AI client.

02 Provide an authorized connection token in the MCP settings so your agent can execute commands securely.

03 Prompt your agent with a specific request, like asking it to retrieve component blueprints and then draft a new story structured as JSON.

The bottom line is that you tell your AI client what content structure you want, and this MCP executes the necessary steps inside Storyblok to make it happen.

Built For

This connector is for content architects, frontend developers, and SEO managers who struggle with manually validating component schemas or updating complex content structures. If your job involves transforming design ideas into production-ready CMS data, you need this.

Content Architect

Uses the MCP to analyze available components and systematically draft new stories, ensuring metadata parameters are correct before handover.

Frontend Developer

Queries component structural IDs or retrieves nested object data programmatically to isolate dependencies for front-end coding.

SEO Manager

Runs checks on existing story structures and bulk updates textual assets across multiple active drafts efficiently.

What Changes When You Connect

01 You can generate a new story or update an existing one using `create_content_story` or `update_content_story`, feeding the agent structured JSON payloads instead of typing out markdown manually.

02 When you need to know what structural parts are available, running `list_components` shows you all schema blueprints. This stops guesswork and standardizes development across teams.

03 To prepare a new campaign page, your agent can first use `list_assets` to find every relevant image or video, making sure the content is sourced before writing begins.

04 Need an overview of what's published? Running `list_stories` gives you titles and metadata for all articles in a space, letting you build a comprehensive index instantly.

05 Before touching content, use `list_space_users` to audit who has access rights. This is critical for maintaining security and controlling modifications across different projects.

Real-World Applications

Launching an Article Draft

A Content Architect needs to publish a new thought leadership piece. They ask their agent to draft it, which uses `list_components` first, then calls `create_content_story`, ensuring the article adheres to the correct format and includes proper metadata.

Updating Product Information

A Marketing Manager updates pricing across fifty product pages. Instead of logging into each one, they instruct their agent to run `list_stories` for all relevant spaces and then use `update_content_story` in bulk.

Building a Component Library

A developer wants to know if the team has already built a 'Testimonial Block.' They ask the agent, which uses `list_components`, confirming the exact structural ID needed for their front-end code.

Media Audit for Compliance

An SEO manager suspects outdated imagery. They tell their agent to run `list_assets` across all spaces, giving them a complete inventory of every media file that needs review or replacement.

Patterns to Avoid

Trying to edit content without context

✗ AVOID

Manually trying to update an article by guessing the correct JSON structure, resulting in errors because you don't know if a field is mandatory or deprecated.

✓ INSTEAD

First, run `list_components` to see the official blueprints. Then use that knowledge when calling `update_content_story` to ensure your data payload matches Storyblok's required schema.

Accidentally deleting core content

✗ AVOID

Running a mass delete command based on an assumption, which permanently removes critical stories or assets without confirmation.

✓ INSTEAD

Always run `list_stories` first to verify the scope of what you're dealing with. If deletion is necessary, use `delete_content_story` only after confirming all necessary data has been backed up.

Forgetting which space to target

✗ AVOID

Running a content update that targets the wrong client environment (Space A instead of Space B), causing massive deployment errors.

✓ INSTEAD

Always start by calling `list_spaces` to get an authoritative list of all environments. Then, specify the exact Space ID when using any creation or retrieval tool.

The Right Fit

Use this MCP if your job requires manipulating content data in a structured way—if you are dealing with component blueprints, asset libraries, or large batches of articles. It excels at programmatic schema generation and validation. Don't use it if all you need is simple drafting or basic text generation; for that, a generic LLM agent will suffice. However, if your goal is to *publish* the content into Storyblok's system, this is mandatory. If you are only interested in user permissions, rely on `list_space_users` instead of trying to guess which API calls might work.

The headache of manual CMS management

Today, managing content means jumping between the visual editor and a separate data panel. You have to copy component IDs here, paste metadata names there, and cross-reference asset URLs in a spreadsheet just to create one page draft.

With this MCP, you simply tell your agent, 'Build me an article about X.' Your AI client handles all the underlying steps—it finds the right blueprints, validates the structure, and writes the payload. You get content that's ready for publication, not a half-finished mess requiring manual cleanup.

Crafting Content with Storyblok MCP

The repetitive, frustrating tasks of listing every asset via `list_assets` or manually checking which components are available using `list_components` disappear. Your agent handles the discovery and validation steps automatically.

What changes is that content becomes a data resource you can manipulate with plain language instructions. You stop clicking; you start commanding.

Storyblok: 9 Content Management Tools

Use these tools to programmatically list, create, modify, and analyze every structural element within your Storyblok content space.

#	TOOL	DESCRIPTION
01	`create_content_story`	Builds a brand new content article by providing its name, slug, and structured JSON data.
02	`delete_content_story`	Permanently removes an existing story from the content repository. This action can't be undone.
03	`get_story_details`	Fetches all specific details for a single, identified content article.
04	`list_assets`	Scans and retrieves a list of every media file stored within the Storyblok space.
05	`list_components`	Shows all structural blueprints that can be used to build content pages.
06	`list_spaces`	Lists every distinct and accessible Storyblok environment or 'space'.
07	`list_stories`	Retrieves a list of article titles and metadata within a specific content space.
08	`list_space_users`	Lists all users who have been granted access permissions for the targeted space.
09	`update_content_story`	Modifies specific fields of an article that already exists in the system.

See It in Action

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

U List the recent articles from my Storyblok space and detail their structural components.

Connected to your Storyblok space.
Recent Stories (`list_stories` **)**:
- `Introduction to LLMs`
- `Marketing Automation Update`
- `Platform Change-log`

Inspecting component structure via `get_story_details`:
- Components: `hero_banner`, `rich_text`, `call_to_action`. Ready for modifications.

U List the structure blueprints by calling list_components and then formulate a new JSON to create a blog story.

Analyzing current schemas...

Available Components (`list_components` **)**:
- `article_body`
- `media_header`

Structuring the payload...
Created Story (`create_content_story` **)**:
Successfully published the new blog story 'Future Tech' into the content repository.

U List all multimedia assets in my Storyblok space and display their URLs.

Scanning the media repository...

Assets Discovered (`list_assets`):
- `product_demo.mp4` → `//a.storyblok.com/.../demo.mp4`
- `hero_bg.jpg` → `//a.storyblok.com/.../hero.jpg`

These assets are ready to be integrated into your active story components.

Frequently Asked Questions

01 **How do I find out which Storyblok spaces are available using the Storyblok MCP?**

You run `list_spaces`. This tool gives your agent a list of all accessible environments, letting you target the right project before any work begins.

02 **Can I use the Storyblok MCP to check if an article exists?**

Yes. You can first run `list_stories` within a specific space ID. This gives you a list of existing titles, confirming if your desired content piece is already published.

03 **How does the Storyblok MCP help with structured data creation?**

You use `create_content_story`. You just provide the name and the JSON payload, and it handles building the article according to your specified schema.

04 **What if I need to modify an existing blog post using Storyblok MCP?**

Use `update_content_story`. This requires you to provide both the space ID and the story ID, plus the specific fields you want to change in the article.

05 **Does the Storyblok MCP let me see what media is available?**

Yes. The `list_assets` tool scans your entire media library and gives you a full list of all images, videos, and other files ready for integration into your content.

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 `"storyblok": { "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

Storyblok 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 Storyblok. 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	Storyblok MCP
Server ID	`019d760d-9bca-70b9-8b9b-59bcf6ebadf5`
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/storyblok.