MCP SERVER | NO CODE | CLOUD HOSTED

Builder.io MCP for AI Agents

Manage Content and Assets in a Headless CMS

Builder.io gives your AI agents full control over your visual CMS and headless content. You fetch entries, update model data, execute complex GraphQL queries, and manage assets—all from a single conversation with your favorite AI client.

C Quality Score 71.43/100

visual-cms | graphql | content-api | asset-management | digital-experience

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/builderio-visual-cms

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.

Builder.io (Visual CMS) MCP

9 tools available
Cloud-hosted on Vinkius

Managing a modern digital experience means coordinating content across multiple systems. Instead of jumping between the Builder dashboard and separate API tools, this MCP connects your entire visual CMS directly to your agent. You can use natural language instructions to interact with your headless content models. Need to check if a specific landing page has been updated? Your agent runs a query against the Content API. Want to push out five new blog posts? You tell your agent to create or update those records programmatically, bypassing the manual UI steps entirely. This integration lets you treat your entire CMS—from core content models to media assets—like another function within your workflow. Connecting through Vinkius means all your existing AI clients get immediate access to this full suite of content tools.

Core Capabilities

01 — Read specific content entries

Your agent retrieves data for any specified model, allowing you to see current drafts or published versions.

02 — Get pre-rendered HTML output

It fetches the final, usable HTML code for components, letting you inspect how content will look before deployment.

03 — Create and modify content records

You instruct your agent to write new content entries or update existing models with a simple command.

04 — Execute advanced graph queries

The agent runs complex GraphQL operations against the full Content API, handling detailed data structures and admin tasks.

05 — Manage digital assets

You can tell your agent to upload new images or videos, or delete old ones by their URL.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/builderio-visual-cms** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Builder.io Public Key (and Private Key if you need write access).

02 Your agent connects using the provided keys, establishing a direct pipeline between your AI client and your CMS.

03 You simply ask your agent natural language questions—for example, 'Give me the latest five blog posts'—and it executes the necessary API calls.

The bottom line is that you manage content directly through conversation, without ever leaving your AI chat interface.

Built For

This MCP is for developers and marketers who hate context switching. If your job involves frequently checking CMS data or running bulk updates across different models, this saves hours of clicking through multiple dashboards.

Developer

You query content models directly from your IDE to validate API endpoints without needing to switch contexts.

Content Manager

You automate bulk updates, like changing a global banner message across dozens of pages, using natural language prompts.

Marketing Specialist

You quickly inspect pre-rendered HTML and content targeting to verify SEO readiness for campaigns before launch day.

What Changes When You Connect

01 Stop switching between tools. You keep all your content operations—querying, updating, uploading assets—inside the chat window.

02 Verify complex component layouts instantly. Use the `get_html` function to pull pre-rendered HTML for SEO checks or client review.

03 Automate data migrations and bulk writes. Need to change a field across 50 models? Your agent handles it with one prompt, using tools like `update_content`.

04 Full GraphQL access means no guesswork. You execute precise queries via `query_graphql`, going deeper than the basic content fetching functions.

05 Maintain clean digital assets by giving your agent permission to both upload new media (`upload_asset`) and delete obsolete files (`delete_asset_by_url`).

Real-World Applications

Auditing a global campaign's landing pages

A marketing manager needs to check if the 'Sale Ends Soon' banner is live on 20 different regional pages. Instead of opening 20 tabs, they prompt their agent: 'Check the rendered HTML for all content entries in the 'banner' model and confirm the current sale date.' The agent uses `get_html` and reports back instantly.

Debugging a multi-model data dependency

A developer finds that blog posts aren't linking to the correct author profile. They ask their agent to run a complex GraphQL query via `query_graphql` across both 'article' and 'author' models, immediately pinpointing the broken relationship in the database structure.

Responding to urgent content changes

A PR crisis requires changing a temporary notice site-wide. The operations team instructs their agent: 'Update all entries in the 'announcement' model with ID X and Y to read 'New messaging.''. This uses `update_content` for immediate, controlled deployment.

Onboarding a new content contributor

A copywriter needs to test how their text looks across three different component types. They use the agent to call `get_html` on sample components, receiving instant, verifiable HTML output for quick sign-off.

Patterns to Avoid

Treating content data like a spreadsheet

✗ AVOID

Trying to manually input every field value into the agent and hoping it gets written correctly. This is slow, error-prone, and doesn't handle relationships.

✓ INSTEAD

Use `create_content` or `update_content`. Instead of listing fields, describe the *outcome*. Say: 'Create a new post about X for model Y.' The agent handles the schema mapping.

Assuming all content is visible via simple search

✗ AVOID

Asking the agent simply to 'List all my content' and expecting everything. CMS models have complex relationships, and basic calls miss critical data.

✓ INSTEAD

Use `query_graphql`. This tool lets you define exactly which fields, relationships, and filters you need, ensuring you get a complete picture of your structured data.

Over-relying on the visual builder for everything

✗ AVOID

Sticking to the Builder dashboard even when running batch operations. The UI is great for humans, but terrible for large-scale automation.

✓ INSTEAD

Use `admin_graphql`. When you need programmatic control—like bulk deletions or system-level changes—the Admin GraphQL tool gives you the necessary power.

The Right Fit

You should use this MCP if your workflow requires treating content data like a service endpoint. If you need to read, write, delete, or query structured CMS content using natural language prompts, connect it. However, don't rely on it for simple drafting; that's still a human job. Also, note its boundaries: while `delete_content` and `admin_graphql` are powerful, they require private keys, so ensure your agent handling is secure. If you only need to manage user permissions or billing data, this MCP won't help—you'll need an identity management integration instead.

Builder.io Content Management with AI Agents

Today, managing content means juggling five tabs: the visual builder, the asset manager, the model structure documentation, the API docs, and your local IDE. You spend minutes copy-pasting data from one place to another just to verify a headline or update an image URL.

With this MCP, you simply ask your agent about the content status. The agent executes the necessary calls using tools like `get_content` or `query_graphql`, compiling a single, actionable answer for you. You get immediate data integrity without opening a single tab.

Builder.io Asset and Media Handling with AI Agents

Manual asset management involves downloading a local file, renaming it according to company standards, and then re-uploading it via the CMS dashboard—a tedious cycle that introduces version control risk.

Now, you tell your agent the asset URL or description. It handles the entire lifecycle: whether you need `upload_asset` for a new hero image or `delete_asset_by_url` to clean up old files, it manages your digital library completely.

9 Tools for Visual CMS Content & Asset Management

Use these tools to fetch data, modify entries, run complex queries, or upload media into your Builder.io environment.

#	TOOL	DESCRIPTION
01	`admin_graphql`	Requires a private key. Executes any administrative GraphQL query or mutation against the CMS.
02	`create_content`	Requires a private key. Builds and publishes an entirely new content entry into a model.
03	`delete_asset_by_url`	Removes media assets from your digital library using only the asset's URL.
04	`delete_content`	Requires a private key. Completely removes an existing content entry or model record.
05	`get_content`	Retrieves specific data entries from any defined Builder.io content model.
06	`get_html`	Generates and returns the final, ready-to-use HTML markup for a given component or model.
07	`query_graphql`	Runs flexible GraphQL queries to pull specific data points from your content structure.
08	`update_content`	Requires a private key. Modifies the existing fields and data of an already published content entry.
09	`upload_asset`	Requires a private key. Uploads new media files, such as images or videos, to your asset library.

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 latest 3 articles published in the 'finance' model? Can you give me their IDs and author names?

Latest Articles Found

ID	TITLE	AUTHOR
abc-123	Q4 Earnings Report Analysis	Jane Doe
def-456	Crypto Market Prediction for 2024	John Smith
ghi-789	Best Practices in Tax Writeoffs	Sarah Lee

I pulled this data using the Content API. Let me know if you want the full text body for any of these.

U I need to update a component on the homepage. The image URL is broken, and I also want to change the main headline.

✅ **Updates Complete**

1. **Image Updated:** Successfully uploaded the new hero shot (URL: `cdn.builder.io/image/...`).
2. **Content Patch:** The 'Homepage Header' entry was updated with the text: 'New Service Launching Next Month!'

The change is live in your Builder space.

U Run a query to find all content models that reference assets from the old marketing folder.

Query Results:

I found 14 entries referencing assets tagged 'old_marketing'. Three of these are in the 'Case Study' model, and eleven are in the 'About Us' section. Would you like me to list the specific asset IDs or delete those references?

Frequently Asked Questions

01 **How can I use Builder.io with my AI agents using the Builder.io MCP?**

You connect your preferred AI client and give it permission to access your CMS data. This lets you manage content, read entries, and execute complex queries directly through natural conversation.

02 **Do I need to know GraphQL to use the Builder.io MCP?**

No, you don't. While the tool uses GraphQL internally for advanced querying (`query_graphql`), you just ask your agent in plain English what data you need.

03 **Can this MCP help me with SEO checks on my content?**

Yes. You can use the MCP to retrieve pre-rendered HTML markup for specific components, which is essential for manually checking how search engines will view your finished page.

04 **What if I need to make bulk changes across multiple pages in Builder.io?**

You can instruct the agent to run batch operations using tools like `update_content`. This lets you change a global element, such as a footer disclaimer, on dozens of models at once.

05 **Is this MCP only for content, or can I manage media files too?**

It handles both. You can upload new images and videos using `upload_asset` and even delete old assets by their URL to keep your digital library clean.

06 **Can my AI agent connect this MCP to other tools I use?**

Yes, because it's hosted on Vinkius, your agent gains access to the entire catalog. This means you can link content data from Builder.io with inputs from other specialized services.

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 `"builderio-visual-cms": { "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

Builder.io (Visual CMS) 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 Builder.io (Visual CMS). 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	Builder.io (Visual CMS) MCP
Server ID	`019e3871-ac1a-719e-b0e3-13e31f3a6d86`
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/builderio-visual-cms.