

MCP SERVER

NO CODE

CLOUD HOSTED

Builder MCP for AI Agents

Automating Content Block Generation and Model Updates

Builder MCP gives your AI client full programmatic control over any headless CMS workflow. It lets you generate, update, and manage structured content blocks, inspect data schemas, and orchestrate visual components directly from natural conversation.

D Quality Score 55/100

visual-editor

digital-experience

content-automation

schema-management

web-development

no-code



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

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

FinOps

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

Firewall & DLP

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

Agent Activity

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

Tool Health

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

Incident Log

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

Get started at [cloud.vinkius.com](https://vinkius.com) — connect your AI agent in under 60 seconds.

Builder MCP

10 tools available

Cloud-hosted on Vinkius

Managing a complex website's content usually means jumping between the CMS UI, running local schema checks, and manually updating component models. This MCP changes that. You connect your Builder space to any AI agent and treat your entire headless CMS architecture like an API call—you talk to it, and it acts.

You can ask your agent to list every available data model or check the exact JSON structure needed for a new content block. Need to update 50 product descriptions? Just tell your agent to do it. Want to delete old, unused page components? It handles that too. This isn't just another API wrapper; it's an operational layer that lets you treat content creation and maintenance as conversational tasks. Connecting this MCP via Vinkius means you get the full Builder catalog accessible from any compatible AI client, keeping your coding focused on logic, not repetitive CMS administration.

Core Capabilities

01 — List available data models

Retrieve a list of all defined content models and schemas within your builder space.

03 — Find content records

Search the CMS to pull specific content documents or count how many items exist in a given model.

05 — Manage media assets

Check metadata and URLs for uploaded images and files stored on the Builder platform.

02 — Inspect model structure

Get the precise field definitions and JSON boundaries for any specific content model you name.

04 — Create and modify content blocks

Generate entirely new visual components, or update existing content entries using precise data inputs.

06 — Remove deprecated elements

Permanently delete specific content entries or old components from your live CMS models.

One Click on Vinkius — From Prompt to Execution

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

- 01** Subscribe to this MCP and input your Builder Public and Private API Key pair.
- 02** Your AI client fetches the available schemas, allowing you to read the data structure right inside your chat environment.
- 03** You prompt the agent with a content request—like 'List all blog post models' or 'Update the hero section title on page X.'—and get instant action.

The bottom line is that you use natural language prompts to perform complex, multi-step CMS operations without writing a single function call.

Built For

This MCP is essential for developers and content teams who interact with headless CMS systems daily. It solves the friction of context switching between your code editor, documentation pages, and the visual builder interface.

Frontend Developer

Needs to retrieve the exact data schema for a Builder model before writing frontend fetch logic.

Content Editor / Manager

Must automate content tasks, like pushing multiple translated copies of a landing page section automatically.

DevOps Engineer

Requires the ability to wipe orphaned elements or test environment components without touching the UI dashboard.

What Changes When You Connect

- 01** You instantly get the full schema structure of a model using 'get_model_schema,' eliminating guesswork before writing frontend code.

-
- 02 Never manually update content again. You can push 50 translated copies of a page section or landing block automatically using 'create_visual_block' and 'update_visual_block'.

 - 03 Need to find a specific piece of content? Use 'get_single_content' with query strings instead of clicking through endless pages.

 - 04 'list_builder_models' gives you an immediate, comprehensive map of every data structure in your CMS, saving discovery time.

 - 05 DevOps can use 'wipe_visual_block' to clean up orphaned components and test elements without risking the live UI.

 - 06 The system tracks media assets through 'get_media_file,' giving you all necessary URLs and metadata in one place.
-

Real-World Applications

A developer needs to validate a new component's data structure

Instead of guessing the required fields, the developer asks their agent to 'get_model_schema' for the target model. The agent immediately returns the strict JSON definitions, allowing them to write type-safe code instantly.

A site administrator needs to clean up old test data

The admin tells their agent to 'wipe_visual_block' for components tagged as deprecated. The agent executes the deletion and reports back on the number of elements removed, keeping the CMS clean.

A marketing team needs to refresh multiple page sections

The editor asks their agent to 'update_visual_block' across several specific content entries. The agent executes the update and confirms all 15 required fields were changed, saving hours of manual API calls.

A team wants to check content volume before a launch

The lead asks their agent to 'count_model_entities' for the product catalog model. The agent quickly confirms there are 4,201 active items ready for review.

Patterns to Avoid

Trying to fetch data without knowing the schema

X AVOID

The developer tries to write a query assuming a field exists (e.g., `product_color`) but it's actually named `variant_hue`. The API call fails, causing debugging delays.

✓ INSTEAD

First, use 'get_model_schema' to read the exact data model structure. Then, reference those confirmed fields in your query using 'get_single_content'.

Manually listing models one by one

X AVOID

The user has 12 different content models and must run a separate command for each one just to see what exists, wasting time.

✓ INSTEAD

Use 'list_builder_models' first. This gives you an instant inventory of all available data types in the entire Builder space.

Confusing content listing with model definition

X AVOID

The user asks to list all blocks, but doesn't specify which model they belong to, resulting in ambiguous or empty results.

✓ INSTEAD

Always use 'list_model_content' and ensure you name the specific model (e.g., 'blog-post') that contains the content you need.

The Right Fit

Use this MCP if your workflow involves repetitive, structured interaction with a headless CMS—specifically generating new data, modifying existing blocks, or inspecting schema definitions. It's built for automation around the core CMS architecture.

Don't use it if you are simply trying to read unstructured text content (e.g., full articles). For that, you need a general document retrieval tool. Also, if your primary goal is just simple asset storage and linking, a dedicated cloud storage MCP might be better. This MCP shines when the *structure* of the data matters as much as the data itself.

Builder MCP: Automating Headless CMS Content Blocks

Today, updating content is tedious. You have to open the CMS dashboard, navigate through models, click into a specific page section, and manually update every field—title, subtitle, link, image URL. This process requires constant context switching and makes large-scale rollouts slow.

With this MCP, you simply tell your agent what needs changing. The agent handles the navigation; it knows which model to target and executes the 'update_visual_block' call perfectly, confirming that all data fields are handled in one conversational step.

Builder MCP: Managing Complex Model Schemas

Without this connector, developers spend time guessing the correct field names or JSON structure. They might write code that fails because they assumed a model had a field like `user_bio`, when it's actually called `author_summary`.

Now, you just ask your agent to 'get_model_schema.' It instantly provides the definitive source of truth for every data point. You get reliable, predictable structure that lets you build robust and fail-safe frontend logic.

Builder MCP: 10 Tools for Content Model Operations

Use these tools to manage schemas, list content, create new visual components, and update models within your Builder.io environment.

#	TOOL	DESCRIPTION
01	<code>count_model_entities</code>	Quickly counts how many live items are stored within a specified data model.
02	<code>create_visual_block</code>	Generates new content entries or visual blocks inside any designated Builder model.
03	<code>wipe_visual_block</code>	Permanently deletes a specific piece of content from your Builder.io workspace.
04	<code>get_single_content</code>	Retrieves one specific content document by matching it against a query string on the model.
05	<code>get_media_file</code>	Gathers details, including URLs and metadata, about an uploaded media asset in Builder.io.
06	<code>get_model_schema</code>	Retrieves the exact field structure and schema definitions for a single content model.
07	<code>list_model_content</code>	Fetches a list of all dynamic content blocks or pages associated with a specific Builder.io model.
08	<code>list_builder_models</code>	Lists every defined data model and schema available within the entire Builder space.
09	<code>search_active_models</code>	Finds specific Builder models by matching a given criteria or substring name.
10	<code>update_visual_block</code>	Modifies an existing content block or document within the Builder.io CMS.

See It in Action





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

U I need a list of all content models in my workspace.



Builder Model Inventory

We've got four primary data types configured:

-  `page` : Used for full landing pages.
-  `blog-post` : Standard article format with author and date fields.
-  `section` : Reusable visual blocks like testimonials or CTAs.
-  `author` : Stores biographies for contributors.

Which of these do you want me to check the specific field rules for?

U What's the structure for the 'custom-hero' model?



Schema for `custom-hero`

The Builder.io CMS requires these fields for a valid component:

FIELD NAME	DATA TYPE	REQUIRED?
<code>title</code>	Text String	Yes
<code>subtitle</code>	Text String	No
<code>cta_link</code>	URL	Yes
<code>bg_image</code>	Media Asset	No

Looks like you'll need those four to build a new hero component.

U Can you find the home page and check its structure?



I queried for content titled 'Home' on the `page` model. Found one match (ID: d9e72a4...). It is published and contains a complex array of visual blocks.

Do you want me to retrieve the full JSON output? That dump will show every nested element, including data from models like `section` and `custom-hero`, so you can see exactly how it's built.

Frequently Asked Questions

01 How does using the Builder MCP help me automate content updates?

It lets your AI client perform repetitive CMS tasks without you ever opening the dashboard. You can ask it to update 50 pieces of content across multiple models, and it handles the data input and publishing process for you.

02 Do I need to write code every time I change a model structure?

No. With this MCP, your agent can first run `'get_model_schema'` to tell you the exact field definitions. This means you get the necessary data blueprint before writing any front-end logic.

03 Is this only for creating new content?

No. It handles the full lifecycle. You can list all existing content blocks using `'list_model_content,'` update them using `'update_visual_block,'` or permanently remove them with targeted deletion.

04 Does this MCP handle media assets and URLs?

Yes, it does. You can use the built-in tools to inspect metadata and fetch specific URLs for images and other files used in your CMS content.

05 If I want to know what models exist, how do I start with Builder MCP?

You simply tell your agent to list all available builder models. It will return a comprehensive inventory of every data structure configured in your workspace immediately.

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 `"builder": { "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 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](https://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. 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 MCP
Server ID	019d7565-859b-73dc-86b3-bd304fff0b84
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/builder.