MCP SERVER | NO CODE | CLOUD HOSTED

Framer MCP

Automate CMS Content & Site Publishing Via AI

Framer MCP connects your AI agent directly into Framer's backend, letting you manage site content and publishing workflows conversationally. You can list, create, and update CMS collections, query site structures, and trigger full website publishes—all without opening the visual design editor.

A+ Quality Score 100/100

web-design | cms-management | site-publishing | interactive-design | content-sync | visual-canvas

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

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Framer MCP

8 tools available
Cloud-hosted on Vinkius

Manage your entire digital presence from plain language commands. This MCP gives your agent direct control over Framer's Content Management System (CMS) and publishing pipeline. You can ask it to list all available collections or create a new team profile item, ensuring your content is always accurate and up-to-date.

Need to update a blog post across three different collection types? Your agent handles that by pushing changes directly into the CMS. When everything is ready, you simply ask it to publish the site, making those updates visible to visitors immediately. It's about treating your website content like structured data—something you manage with text commands, not clicks in a GUI.

It works as part of the Vinkius catalog, giving your AI client access to Framer alongside thousands of other services. This means you don't need separate tools for every platform; everything connects through one place.

Core Capabilities

01 — Manage Content Collections

List or create new items within any CMS collection, like adding a new employee profile or case study.

02 — Query Site Structure

Get details about the entire project setup, including all available site pages and content model schemas.

03 — Push Content Updates

Programmatically push fresh data or updates into your existing CMS collections from an outside source.

04 — Deploy Website Changes

Trigger a full site publish to ensure all the changes you made—the new blog post, the updated portfolio—go live immediately.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to the Framer integration on Vinkius and generate an API key from your Framer project's settings.

02 Connect your AI client using this MCP, passing it the required API credentials.

03 Tell your agent exactly what you need done—for example, 'Create a new case study for Acme Corp and publish the site.'—and watch it execute the steps.

The bottom line is that your AI client executes complex content workflows using simple natural language requests.

Built For

Anyone who spends time updating or publishing websites manually knows this pain. If you're a marketing manager constantly juggling CMS updates, a developer needing to sync data from external APIs, or a design lead who wants to maintain content without touching the canvas, this MCP is for you.

Content Manager

Updates blog posts and portfolio items by asking the agent to create new collection items and then triggering a publish.

Web Developer

Writes code that programmatically fetches data from an external database and uses the MCP's tools to sync it into Framer's CMS collections.

Digital Marketing Lead

Manages campaign launches by instructing the agent to update several landing pages, then running a publish command to make them live instantly.

What Changes When You Connect

01 Stop logging into Framer manually. You can use the `create_collection_item` tool to add new blog posts or team bios directly through conversation.

02 Need to know what content exists? The `list_collections` and `list_collection_items` tools let your agent query everything, saving you from clicking through endless dashboards.

03 When a site update is ready, the `publish_site` tool makes it live instantly. It bypasses manual deployment steps and pushes changes immediately to visitors.

04 You don't need to touch the canvas. Use this MCP to manage complex data flows, syncing external information into CMS fields using structured commands.

05 Get a complete overview of your site with `list_pages` and `get_site_info`. Your agent can confirm if all necessary pages are configured before you run a publish command.

Real-World Applications

Launching a New Product Line

A marketing lead needs to launch 10 new product pages. Instead of manually creating each item, they ask their agent to 'Create 10 new collection items for the Products category.' The agent uses `create_collection_item` for all ten entries, then runs `publish_site` so the entire catalog goes live simultaneously.

Updating Team Bios After Reorg

The HR team needs to update 15 employee profiles. They ask their agent to 'Update the bio and role for all staff members in the People collection.' The agent uses `list_collection_items` first, then selectively updates data, finishing by calling `publish_site`.

Syncing Database Content

A developer needs to sync product inventory from an external Postgres database. They configure their client to push this structured data into Framer's CMS collections and then use the MCP to trigger a site publish, making the updated prices visible.

Pre-Flight Site Check

Before launching a major campaign, the content manager asks their agent to 'List all pages and check if the main landing page is configured.' The agent uses `list_pages` and `get_site_info` to confirm site readiness before allowing the publish command.

Patterns to Avoid

Trying to edit design elements

✗ AVOID

Asking the agent, 'Change the color of the main hero background.' The agent will fail because this MCP only handles content data and site publishing, not visual styling.

✓ INSTEAD

If you need content changes, use tools like `create_collection_item` to add text or images. If you need to know what *can* be changed, ask the agent to 'list all CMS collections' first.

Publishing without checking data

✗ AVOID

Running `publish_site` immediately after a manual change, only to find that half of your content is missing or outdated.

✓ INSTEAD

Always check the current state first. Run `list_collection_items` for the specific collections involved, and then run `get_site_info` before executing the final publish command.

The Right Fit

Use this MCP if your core workflow involves managing structured content—things like blog posts, product data, team members—and deploying those changes across a live website. You need an agent to treat Framer as a backend API endpoint, not just a visual editor. Don't use it if you need to perform purely visual design tasks (like adjusting spacing or fonts) because this MCP only manages content and publishing. If your goal is to build complex database logic outside of CMS records, connect that separate data source first; then use the `list_collection_items` tool to push the clean results into Framer.

The Content Workflow Nightmare

Today, updating a single website page often means jumping through three different tabs: the CMS editor to rewrite text; the asset library to find new images; and finally, manually clicking 'Publish'

With this MCP, that process becomes a single conversation. You tell your agent exactly what needs updating—say, changing the mission statement across five different landing pages—and

in the top right corner. If you have 20 pages to update, that’s twenty sets of clicks, logins, and potential human error.

it handles the content updates using `create_collection_item` or similar tools, followed by one command to make everything live.

Content Management with Framer

Manual steps that disappear include navigating the CMS UI, finding which collection an item belongs to, and remembering to check for orphaned content. You no longer have to manually run a site audit every time you change something.

What's different is control. Instead of hoping your human coworker remembers to hit publish, you use `publish_site` via natural language, ensuring the changes are deployed reliably and immediately.

Framer: 8 Tools for CMS & Site Management

Use these tools to list project details, manage site pages, create new content items in your CMS collections, or push updates live from an AI agent.

#	TOOL	DESCRIPTION
01	`list_projects`	Retrieves a list of all Framer projects associated with your account.
02	`get_project`	Fetches detailed configuration and information for a specific Framer project.
03	`list_collections`	Reads and returns the names of all content management collections available on your site.
04	`list_collection_items`	Retrieves a list of existing items (like blog posts) within a specific CMS collection.
05	`create_collection_item`	Generates and adds brand new content items to a specified CMS collection.
06	`publish_site`	Makes all current changes—content, pages, etc.—visible to the public website visitors.
07	`list_pages`	Lists every page defined in your Framer site structure.
08	`get_site_info`	Grabs general configuration data and settings details for the entire website.

See It in Action

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

U List all CMS collections in my Framer site.

Your Framer project has 3 CMS collections: 'Blog Posts' (18 items), 'Team Members' (12 items), and 'Case Studies' (7 items).

U Add a new team member 'Ana Silva' to the Team Members collection.

Team member 'Ana Silva' added to the 'Team Members' collection. Fields set: Name, Role (empty). You can ask me to update additional fields like bio, photo URL, or social links.

U Publish my Framer site with the latest CMS changes.

Site publish triggered. Your CMS changes are now live. Last updated: just now. All 37 collection items synced across 3 collections.

Frequently Asked Questions

01 **How do I get my Framer API key?**

Open your Framer project in the editor. Go to **Site Settings** (gear icon in the top bar), then navigate to the **General** tab. Scroll down to find the **Server API** section and click **Generate API key**. Copy the key and store it securely — treat it like a password. This key is specific to this project. Paste it into the configuration field below.

02 **Can I push content from a spreadsheet into my Framer CMS?**

Yes. Your AI agent can create and update CMS collection items programmatically. Describe the data you want to add, and it maps your fields to the Framer collection schema and creates the items — perfect for bulk content migrations or data-driven landing pages.

03 **Does publishing through the API affect my Framer design?**

No. The Server API only manages CMS content — it cannot modify your visual design, canvas layout, or component structure. Your designers remain in full control of the visual experience while content teams operate independently through the AI agent.

04 **Is this suitable for agencies managing multiple Framer projects?**

Yes. Each API key is bound to a specific Framer project. Configure separate integrations for each client project and your AI agent can switch between them — managing content across your entire portfolio without opening multiple browser tabs.

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 `"framer": { "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

Framer 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 Framer. 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	Framer MCP
Server ID	`019d759e-ac8e-734d-b871-5a1d3f28a486`
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/framer.