

MCP SERVER

NO CODE

CLOUD HOSTED

MJML Email Markup MCP

Convert complex markup into flawless, cross-client HTML.

MJML Email Markup MCP transforms complex MJML code into responsive HTML emails. Stop debugging CSS for different email clients; simply give your agent the markup and get production-ready HTML that looks right everywhere, from Outlook to Gmail.

A+ Quality Score 100/100

email-templates

responsive-design

html-rendering

markup-language

email-development



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 — connect your AI agent in under 60 seconds.

MJML (Email Markup) MCP

1 tools available

Cloud-hosted on Vinkius

Email development is a nightmare because every client—Outlook, Apple Mail, Gmail—treats HTML differently. Manually coding these templates means fighting complex nested tables and worrying about CSS inline conflicts. This MCP solves that by letting your AI agent handle the heavy lifting. You provide MJML markup (XML or JSON), and this connector transpiles it instantly into clean, responsive HTML. It's like having a senior front-end developer available in chat. Whether you're working directly in an IDE or just chatting with your agent, you get immediate, client-standardized results. Because Vinkius hosts this MCP, you connect once and gain access to industry best practices for email design without needing a specialized build system.

Core Capabilities

01 — Generate responsive HTML from markup

Takes raw MJML code strings and converts them into production-ready HTML that adapts to various viewing devices.

03 — Ensure cross-client compatibility

Automatically structures the output to meet best practices for major email service providers, bypassing common client rendering errors.

02 — Rapid email prototyping

Allows you to quickly iterate on full email designs directly within your chat or development environment without leaving your workflow.

04 — Validate complex markup syntax

Checks your input MJML code against industry standards, preventing manual formatting mistakes like improper table nesting or missing required tags.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/mjml-email-markup — connect your AI agent in three steps.

- 01** First, you provide the connector with your raw content—this is either a JSON or XML string of MJML markup.
- 02** The MCP uses the `render_mjml` tool to process this input through its engine, which translates the standardized format into browser-readable code.
- 03** You receive clean, responsive HTML output that follows modern email client best practices.

The bottom line is you skip the debugging steps and get finished HTML ready for deployment.

Built For

This MCP is essential for Marketing Operations professionals, front-end developers, and UX designers who spend too much time fighting with email client bugs instead of building campaigns. If your job involves sending emails through multiple platforms, you need this.

Marketing Automation Specialist

They use this to quickly test how a new campaign layout will look in Outlook or Gmail before the actual send date.

Front-End Developer

They rely on it to generate complex, highly structured email sections without manually writing dozens of nested tables and inline CSS.

UX Designer

They use it to validate MJML syntax and get immediate visual confirmation that their design translates correctly into code.

What Changes When You Connect

- 01** Avoids the 'table hacking' headache. Instead of manually nesting tables to appease Outlook, this MCP handles all client-specific structure automatically when you run `render_mjml`.

-
- 02 Saves hours of debugging time. You stop spending cycles fixing minor CSS bugs across different email platforms and start focusing on content.

 - 03 Allows for rapid prototyping. Need a button or section design? Give your agent the markup, use `render_mjml`, and get an immediate visual preview to approve.

 - 04 Guarantees responsiveness. The output isn't just code; it's structurally sound HTML that adapts cleanly whether viewed on mobile or desktop.

 - 05 Keeps you in flow. You don't need to switch between a separate build tool or IDE plugin; your agent handles the conversion right where you work.
-

Real-World Applications

Client needs an email template for a major product launch.

The developer asks their agent: 'Can you use `render_mjml` to take this MJML code and give me the HTML for our new feature announcement?' The agent returns perfect, cross-platform code immediately, bypassing days of manual CSS work.

Designer needs to validate complex button styles.

The designer wants to see how an advanced call-to-action will render. They ask the agent to generate and then use `render_mjml` on the code, validating that the resulting HTML is clean and accurate.

Marketing team needs to test a flyer conversion.

The marketing specialist pastes their raw JSON markup into the chat and prompts: 'Render this for me.' The agent uses `render_mjml` and provides a working, responsive HTML output for pre-deployment review.

Patterns to Avoid

Writing nested tables manually

✗ AVOID

Copying and pasting raw HTML from a WYSIWYG editor often results in massive, unreadable code blocks that break on mobile devices or get stripped out entirely by Gmail.

✓ INSTEAD

Use the MCP to convert your content via ``render_mjml``. This ensures you are generating structured, clean MJML markup first, which then gets converted into robust HTML.

Relying on pure CSS for email structure

✗ AVOID

Trying to build an entire campaign layout using only modern CSS properties often fails because major mail clients (especially Outlook) don't support them.

✓ INSTEAD

This MCP is built specifically around MJML, which encodes best practices that respect the limitations of legacy email clients. Always let ``render_mjml`` handle the final conversion.

Debugging client-specific bugs

✗ AVOID

Spending half a day debugging why a specific background color looks wrong only when viewed in Apple Mail on an older version.

✓ INSTEAD

The MCP's engine is designed to account for these differences. By generating the final HTML via ``render_mjml``, you get code that already accounts for multiple client behaviors.

The Right Fit

Use this MCP if your primary goal is creating professional, responsive email campaigns intended for a wide audience using various clients (Gmail, Outlook, etc.). If the deliverable *must* be an email and you're dealing with complex layouts or varying client behavior, this tool is mandatory. Don't use it if you are building simple web pages that only need to look good in modern browsers; then standard HTML/CSS tools suffice. Also, don't rely on it for basic text messaging campaigns, as the markup complexity isn't needed. This MCP exists solely to solve the specific pain of email rendering.

The Pain of Email Development

Today, building a single campaign means copying content into multiple tools: you draft in a CMS, check it on your phone, then paste it into an editor that only shows a preview. You end up spending hours fighting nested tables and debugging why background colors or columns look different when viewed through Outlook versus Gmail.

With this MCP, the process shrinks down to one step. Your agent takes your source markup, uses `render_mjml`, and instantly provides you with production-ready HTML that has already been tested against industry best practices.

Generate Responsive HTML via `render_mjml`

The manual steps that vanish are: checking for CSS conflicts, ensuring every column is wrapped in the correct table structure, and manually adjusting padding or spacing to accommodate different client limitations.

Now you simply provide the structured input; the MCP handles all the necessary structural changes. It's immediate, reliable HTML.

MJML (Email Markup) MCP: 1 Tool Available

The `render_mjml` tool allows you to take any MJML markup code string and instantly convert it into responsive, clean HTML.

#	TOOL	DESCRIPTION
01	<code>render_mjml</code>	Takes raw MJML markup (XML or JSON) and renders it instantly to responsive HTML.

See It in Action

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

- U** Render this MJML code to HTML: `<mjml><mj-body><mj-section><mj-column><mj-text>Hello World</mj-text></mj-column></mj-section></mj-body></mjml>`



I've processed your MJML markup. The transpiled responsive HTML is ready and follows all email client best practices. You can view the output in the presenter below.

- U** Can you use `render_mjml` to convert a JSON-based MJML structure into a responsive email?



Absolutely. Please provide the MJML JSON string, and I will use the `render_mjml` action to generate the corresponding high-quality HTML for you.

- U** Generate a responsive button in MJML and render it to HTML.



I've created a responsive button using `<mj-button>`. I am now running `render_mjml` to provide you with the production-ready HTML code and a visual preview.

Frequently Asked Questions

01 Does MJML (Email Markup) MCP support JSON or XML?

Yes, it supports both. The `render_mjml` tool accepts raw markup strings whether they are formatted as XML or JSON.

02 Can I use `render_mjml` to just generate a simple text email?

While you can pass minimal markup, this MCP is optimized for complex, styled emails. For basic text announcements, standard content management systems are faster.

03 Is the HTML output from MJML (Email Markup) MCP production-ready?

Yes, the output is designed to be production-grade, incorporating best practices that solve common client rendering issues. You don't need further structural cleanup.

04 How does render_mjml handle responsiveness?

It handles responsiveness by generating code structures that adapt columns and sections based on the viewing device size, ensuring a good experience across mobile, tablet, and desktop clients.

05 What if my MJML markup has an error? Can render_mjml catch it?







The process is designed to validate your syntax. If there's an issue with the input structure, the agent will alert you before attempting to generate the HTML.

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 <code>"mjml-email-markup": { "url": "..."} </code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

MJML (Email Markup) 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 MJML (Email Markup). 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	MJML (Email Markup) MCP
Server ID	019e38c2-db09-72c6-9310-9ea0e4606917
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/mjml-email-markup.