

MCP SERVER

NO CODE

CLOUD HOSTED

# Mustache Template Engine MCP

Generate reliable documents from structured data.

Mustache Template Engine MCP renders complex documents and emails by filling data into structured templates. You provide a template (using Mustache syntax) and raw JSON data, and the engine deterministically outputs final content—like complete HTML or formatted reports. It works universally across multiple programming languages like JavaScript, Python, and Go.

**F** Quality Score 12.97/100

templating

logic-less

data-binding

code-generation

document-automation



# 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://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# Mustache Template Engine MCP

1 tools available

Cloud-hosted on Vinkius

Generating standardized documents used to be a mess. Manually stitching together email bodies or invoices by concatenating strings is brittle; one missing bracket or misplaced tag breaks the entire layout. Worse, if your code hallucinates a variable name that doesn't exist in the data, your process fails silently or outright.

This MCP solves that problem using Mustache templates. These templates are designed to be logic-less—they only handle slots and data, never executing complex business logic. This means you get predictable output every time. Because it adheres to a universal specification, you can write the same template once and render it identically in JavaScript, Python, or any other language your team uses. You just feed it the raw JSON payload, and this MCP takes care of the rest, giving you clean, finished content right away.

---

## Core Capabilities

### 01 — Generate structured emails

Renders full HTML email bodies by mapping data fields into predefined template slots.

### 03 — Loop through lists of data

Iterates over arrays in your JSON payload, allowing you to generate itemized tables like product line items or user history.

### 05 — Support multiple languages

The engine uses a single standard that guarantees consistent output regardless of whether your backend is written in Python or Java.

### 02 — Process document sections

Handles conditional content rendering (e.g., showing a tax section only if the order total exceeds \$100).

### 04 — Maintain code safety

Because the templates are logic-less, they prevent common security risks associated with executing arbitrary code during rendering.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/mustache-template-engine](https://vinkius.com/mcp/mustache-template-engine) — connect your AI agent in three steps.

- 01 You provide the Mustache template string, which contains placeholders for data, and the structured JSON object containing all necessary variable values.
- 02 The MCP engine parses the template, executing the logic-less rules (like conditional sections or list loops) against your provided data.
- 03 It returns a finalized string—ready to be used as an HTML body, text message, or report section.

The bottom line is that you stop writing complex rendering code and just focus on the data structure needed for the final output.

---

## Built For

Backend engineers, content developers, and DevOps specialists who spend too much time manually testing template outputs or fighting brittle string concatenation in their core services.

### Software Engineer

Uses the MCP to generate standardized output like invoices or confirmation emails from backend data payloads.

### DevOps Specialist

Builds automated notification systems that reliably format complex alerts (like system status changes) into readable formats for Slack or ticketing systems.

### Product Manager

Defines the data requirements and template structure, ensuring the engineering team can generate consistent user-facing documents across different product lines.

---

## What Changes When You Connect

- 01 Reliability: You eliminate the risk of layout breaks. Instead of manually concatenating strings, using `render_mustache` ensures that complex HTML structures remain intact every time.

- 
- 02 Consistency: The universal spec means your templates work identically whether your primary language is Python or JavaScript. One template, guaranteed output across all platforms.

---

  - 03 Safety: Because Mustache is logic-less, the engine prevents code injection risks and stops you from accidentally trying to run server-side logic inside a document template.

---

  - 04 Complexity Handling: You don't need separate code for list items or conditional sections. The engine handles loops ( `{{#section}}` ) and optional elements automatically.

---

  - 05 Speed of Development: Instead of writing dozens of lines of data formatting code, you define the structure once in the template and let your agent handle the rendering.
- 

---

## Real-World Applications

### Generating a Welcome Email

A new user signs up. Instead of building an email body with hardcoded placeholders, the agent calls `render_mustache`, passing the HTML template and the user's JSON data (name, activation link). The result is a perfectly formatted welcome email ready to send.

### Building Automated Alerts

A monitoring system detects an error. The agent uses the template engine to render a detailed Slack notification message, pulling in the severity, service name, and timestamp from the event data payload.

### Creating an Invoice Document

The billing service needs to generate a PDF body. By providing the template with line item loops and total calculations in JSON format, calling `render_mustache` creates a fully itemized, mathematically correct invoice ready for printing.

---

# Patterns to Avoid

---

## Manual string concatenation

### ✗ AVOID

Writing code like ``header + "<p>" + item.name + "</p>"`` to build a list of items, which inevitably breaks when a field contains an apostrophe or the structure changes.

### ✓ INSTEAD

Use the dedicated tool. Pass the template and data payload to ``render_mustache``. This handles all escaping and structural requirements automatically.

---

## Using generic HTML templating

### ✗ AVOID

Relying on general-purpose text processing libraries that might execute code or struggle with nested loops, leading to unpredictable output.

### ✓ INSTEAD

Stick to the Mustache spec. This MCP enforces a safe, universal standard that only allows data insertion and basic structural control.

---

## Hardcoding sections

### ✗ AVOID

If the document needs an optional 'Discount Code' section, you have to write ``if (data.discount) { addSection(); }`` every time.

### ✓ INSTEAD

Use Mustache's built-in conditional syntax in your template and let the engine handle the omission or inclusion of the entire block via ``render_mustache``.

---

## The Right Fit

Use this MCP if your goal is to take structured data (like a list of products, user details, or financial totals) and consistently format it into a predefined output structure—whether that's HTML, plain text, or XML. This engine excels at generating standardized documents like emails, receipts, and reports.

Don't use this if you are trying to generate free-form creative content (e.g., 'Write me a poem about rain') or perform complex business logic that requires external API calls within the template itself. For those tasks, you need an agent capable of reasoning. If your output must be perfectly structured and derived solely from inputs, this MCP is your tool.

---

## The pain of building documents with code

Right now, every time a user needs something complex—like an invoice or a confirmation letter—your team has to write specialized rendering logic. This involves dozens of lines of conditional checks, list loops, and error handling just to stitch together the document body from JSON data.

With this MCP, you define the template structure once using Mustache syntax. You simply feed your agent the raw data payload, and it calls `render_mustache`. The result is a clean, finished document without writing any complex rendering code.

---

## Get reliable documents with `render_mustache`

You eliminate the need for brittle string concatenation and platform-specific templating languages. No more worrying about if Python handles escaped quotes differently than JavaScript does.

This MCP guarantees that your templates are logic-less, stable, and universally compatible. You gain predictable, high-quality output every single time.

---

# Mustache Template Engine: 1 Tool

Use the available tools to render logic-less templates, turning structured data and placeholders into finalized content for emails, reports, and notifications.

#	TOOL	DESCRIPTION
01	<code>render_mustache</code>	Passes a Mustache template string and a JSON data object to deterministically render the final content, avoiding manual string errors.

---

## See It in Action

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

- U** Render this welcome email template with the new user's data: name, email, and activation link.



HTML email rendered with all `{{name}}`, `{{email}}`, `{{activationLink}}` placeholders replaced.

- U** Generate an invoice PDF body with line items, subtotal, tax, and total from this JSON order.



Invoice template rendered with itemized list, calculated totals, and formatted currency.

- U** Create a Slack notification message from this alert template and the monitoring event data.



Slack message rendered with severity, service name, error message, and timestamp.

---

## Frequently Asked Questions

### 01 What is the difference between Mustache Template Engine MCP and simple data binding?

Simple data binding just replaces variables (e.g., `{{name}}`). This MCP handles complex structure, including conditional sections and loops over arrays, letting you generate entire lists of itemized content.

### 02 Can I use Mustache Template Engine MCP to render PDFs?

While it doesn't output a PDF file directly, it renders the clean HTML or structured text body that you can then easily pass to a separate service for final PDF conversion.

---

**03 Does Mustache Template Engine MCP support multiple data types?**

Yes. It handles standard JSON primitives (strings, numbers, booleans) and complex arrays/objects used in loops and conditional rendering.

---

**04 How do I use the render\_mustache tool?**

You must provide two inputs: the template string containing your placeholders, and a corresponding JSON data object. The engine then processes both to generate the final output.

---

**05 Is Mustache Template Engine MCP safe from code injection?**

Yes. Because the templates are designed to be logic-less, they can only insert data; they cannot execute arbitrary code, making them inherently safer than full programming language templating systems.







---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"mustache-template-engine": {   "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	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

# Mustache Template Engine 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](https://vinkius.com) · [support@vinkius.com](mailto: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 Mustache Template Engine. 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	Mustache Template Engine MCP
Server ID	019e38c4-42e5-722e-a53a-4229ddb81e66
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### 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/mustache-template-engine](https://vinkius.com/mcp/mustache-template-engine).