MCP SERVER | NO CODE | CLOUD HOSTED

# Handlebars Template Compiler MCP

## Stop AI from hallucinating your email variables.

Handlebars Template Compiler uses the industry-standard Handlebars engine to flawlessly compile dynamic templates. Instead of asking your AI client to manually search and replace variables or process complex HTML logic, this MCP compiles raw Handlebars code against structured data payloads. It handles loops (like product lists) and conditional statements, guaranteeing production-ready email content and web payload structures every time.

F Quality Score 3.6/100

handlebars | template | email | html | marketing

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/handlebars-template-compiler

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.

# Handlebars Template Compiler MCP

1 tools available
Cloud-hosted on Vinkius

Drafting marketing emails or building dynamic web pages often means dealing with template languages like Handlebars. These templates contain placeholders—things like `{{customer_name}}` or logic blocks ( `{{#if is_premium}}` ). When an AI agent tries to process this, it frequently fails, deleting parts of the HTML or misinterpreting complex loops. This MCP fixes that problem completely. It uses a dedicated compiler engine to merge your raw template code and structured data payloads exactly how they should be. You provide the skeleton (the template) and the ingredients (the JSON data), and we give you perfect, functional HTML output. Because this logic is handled by an established standard, not fuzzy AI inference, you get reliable results for everything from welcome emails to complex product list layouts. When your agent connects to Vinkius, it gains access to this compiler, ensuring that template rendering is always accurate.

## Core Capabilities

**01 — Compile Templates**

The MCP takes a raw Handlebars template string and a JSON data string to generate finished HTML or text payloads.

**02 — Handle Conditional Logic**

It processes complex logic blocks, such as showing content only if a user has a specific status, using the `{{#if}}` syntax.

**03 — Iterate Over Data Lists**

The MCP loops through arrays of data (like product lists) to generate repeatable HTML structures using the `{{#each}}` syntax.

## One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/handlebars-template-compiler** — connect your AI agent in three steps.

01 You pass two pieces of information: your raw Handlebars template containing placeholders, and a separate JSON string that holds all the data (names, prices, etc.).

02 The MCP runs this input through the established Handlebars engine, which correctly maps the JSON values into the template structure.

03 You get back clean, final HTML or text content ready to be used in an email client or web payload.

The bottom line is you stop relying on your AI agent's ability to write code and start using a dedicated tool that guarantees perfect compilation results every time.

## Built For

This MCP is for anyone building marketing automation or web content who gets frustrated when their AI client mangles HTML templates. If you're constantly debugging why a loop failed or why an email variable disappeared, this is your fix.

**Marketing Automation Specialist**

They use this MCP to ensure dynamic welcome emails and promotional campaigns compile correctly before deployment, preventing embarrassing content errors.

**Full-Stack Developer**

They connect this MCP into their agent workflow to reliably render complex data structures for front-end payloads without writing boilerplate compilation code.

**Technical Content Writer**

They rely on it to take structured product feeds and automatically generate thousands of variable, yet perfectly formatted, marketing copy blocks.

## What Changes When You Connect

01 Flawless template compilation: You stop worrying about variable replacement failures. The `compile_template` tool uses the industry-standard Handlebars engine to guarantee placeholders resolve correctly.

02 Reliable logic handling: It processes complex features like looping through product arrays ( `{{#each}}` ) and showing content conditionally ( `{{#if}}` ), turning raw JSON into solid HTML structure.

03 Guaranteed output format: This MCP doesn't guess; it compiles. You get clean, production-ready code for marketing emails or data payloads that will actually work in the destination system.

04 Fewer debugging headaches: Since template logic is handled by a dedicated compiler, you spend less time fixing syntax errors and more time writing good copy.

05 Handles structured complexity: It easily takes large JSON objects—like an entire product catalog—and injects them into templates meant for individual items.

## Real-World Applications

**Building a Product Showcase Email**

A marketer needs to send an email that shows 3-5 recommended products based on user data. Instead of asking the agent to manually format HTML loops, they use this MCP's `compile_template` tool. They pass the product list JSON and the template containing `{{#each}}`, and the final, correct HTML is generated instantly.

**Creating Dynamic Welcome Pages**

A developer needs a landing page that shows different content blocks based on whether the user subscribed via 'Trial' or 'Paid'. They use `compile_template` to compile a template with an `{{#if}}` block, feeding it the user status data from JSON. The result is perfectly structured HTML.

**Generating Customized Onboarding Payloads**

An Ops Specialist needs to generate personalized onboarding payloads for new hires. They use this MCP to feed a template with placeholders like `{{department}}` and `{{manager_name}}`, ensuring the final data payload is perfectly structured before passing it to the main application.

**Processing Variable Product Feeds**

A technical writer wants to write an email that includes a list of related items. They use `compile_template` with the product feed JSON and the loop logic, ensuring that every single item in the source data gets its own correctly formatted HTML card.

# Patterns to Avoid

### Asking the agent to replace strings

✗ AVOID

The prompt says: 'Please put the name into the template and wrap all prices in <b> tags.' The AI client tries simple text replacement, resulting in broken HTML or missing variables.

✓ INSTEAD

Pass the raw template and the JSON data using `compile_template`. Let the dedicated compiler handle the structured replacement of variables like `{{name}}` and complex logic blocks.

### Attempting to build loops manually

✗ AVOID

A user tries to write a multi-line prompt describing how product lists should be formatted, making the AI client treat the list as mere text instead of structured HTML.

✓ INSTEAD

Use `compile_template`. The tool is designed specifically for this. It correctly interprets template logic like `{{#each}}` and builds proper, repeating HTML elements from your JSON array.

### Relying on the AI to maintain context

✗ AVOID

The agent fails when the data payload is large or contains edge cases (like null values), causing parts of the final email body to be deleted or left blank.

✓ INSTEAD

Use `compile_template`. It executes against a proven engine, providing predictable and reliable compilation results regardless of how complex your source JSON data becomes.

# The Right Fit

You use this MCP if your primary problem is rendering template logic. Specifically, you need to take structured inputs (JSON) and inject them into a predefined markup skeleton (Handlebars template), ensuring that all loops ( `{{#each}}` ) and conditionals ( `{{#if}}` ) work perfectly every time. Don't use it if you just need

simple text variable substitution; your agent might handle that fine. However, do not use this MCP if your task is simply data validation or schema enforcement—for that, you'd want a dedicated Pydantic AI tool. This compiler lives in the middle: it takes clean JSON and turns it into perfectly structured code output.

## The Pain of Manual Content Assembly

Today, compiling dynamic content is a nightmare of copy-pasting and debugging. You write your template in one spot, but when you need to inject product data or user status logic, the AI agent either breaks the HTML with misplaced tags or simply deletes entire sections because it doesn't understand how Handlebars loops work.

It feels like you're constantly babysitting the output, fixing broken `{{#each}}` blocks and manually ensuring every variable resolves. You spend more time debugging template syntax than actually writing marketing copy.

## Compile Templates with Handlebars Template Compiler

With this MCP, you stop worrying about the mechanics of compilation. The `compile_template` tool takes your raw template and your product JSON feed. It executes the full logic—loops, conditionals, all variables—in one reliable step.

The output is clean, structured, production-ready HTML. You get to focus on what you want to say, not how the underlying code has to be assembled.

# Handlebars Template Compiler: 1 Tool

This tool lets you take a raw template and structured data payload, compiling them into final HTML output using industry-standard logic.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `compile_template` | This tool compiles Handlebars templates by taking a raw template string and a JSON data string to generate final HTML content, correctly handling loops and conditionals. |

# See It in Action

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

**U** Compile the template `Welcome {{name}}!` using this data: `{"name":"Alice"}`.

Compiled Template Result: Successfully compiled.

**U** Inject this CRM product list JSON into my Handlebars marketing email.

Compiled Template Result: Output generated.

**U** Render this HTML snippet replacing the `{{url}}` variable.

Compiled Template Result: Success.

# Frequently Asked Questions

01 **How do I use Handlebars Template Compiler with product data?**

You pass two inputs: your template string and a JSON array containing all the products. The `compile_template` tool uses the `{{#each}}` syntax to iterate over every item, generating repeatable HTML for your list.

02 **Can Handlebars Template Compiler handle simple variable replacement?**

Yes, it handles that, but it's designed for more. It uses the `compile_template` function to correctly replace placeholders like `{{name}}`, even if they are nested within complex logic blocks.

03 **What is required in the JSON data for compile_template?**

The JSON must contain all the data points referenced in your template. For example, if your template uses `{{price}}`, the associated JSON object needs a 'price' key.

04 **Does Handlebars Template Compiler only work for emails?**

No. While excellent for marketing emails, it compiles any standard Handlebars template. You can use it to generate payloads or structured HTML for web components too.

05 **Is this better than just asking my AI client to do the compilation?**

Absolutely. This MCP uses an established engine; your agent uses pattern matching. Using `compile_template` prevents the common failures and hallucination that happen when relying solely on LLM inference for code execution.

# 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 `"handlebars-template-compiler": { "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

# Handlebars Template Compiler 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 Handlebars Template Compiler. 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 | Handlebars Template Compiler MCP |
| Server ID | `019e38a5-6193-70f3-846d-ac3030bb781f` |
| 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/handlebars-template-compiler.