MCP SERVER | NO CODE | CLOUD HOSTED

Treblle MCP

Log Every Request and Response in One Place

Treblle connects your AI agent to an API observability platform. It lets you send full API request and response payloads directly into a central dashboard for real-time monitoring. You track performance, debug errors, and document endpoints without manual setup. Stop losing context when debugging; get instant visibility into every API call.

A+ Quality Score 100/100

api-monitoring | observability | api-analytics | request-logging | api-documentation

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

SCAN TO VIEW

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.

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.

Treblle MCP

1 tools available
Cloud-hosted on Vinkius

When your code talks to external services, that conversation generates massive amounts of data—logs, headers, payloads. Sending all that raw traffic manually is a nightmare. This MCP changes that. You connect your agent and it lets you push the entire lifecycle of an API interaction directly into Treblle. Your agent handles sending both the original request and the full response body automatically.

You don't just log; you get observability. As data streams in, Treblle tracks performance metrics and spots errors instantly. Plus, security is built-in: it masks sensitive fields like credit card numbers before anything gets saved. This means your team can focus on debugging and documentation, not worrying about compliance or lost context. When you connect this through Vinkius, you're giving your AI client a single point of truth for all your API traffic.

This is crucial for developers needing a reliable record of what worked, what failed, and how fast it was.

Core Capabilities

01 — Log Full Request/Response Payloads

Send the entire raw body of an API interaction—both the request sent out and the response received back—to Treblle.

02 — Mask Sensitive Data Automatically

Treblle automatically scrubs private information, like passwords or SSNs, from the data before logging it, keeping your system secure.

03 — Attach Custom Context IDs

Add specific identifiers, such as user IDs or trace IDs, to log entries so you can filter and group related calls later.

04 — Monitor Live API Performance

Observe API health and traffic patterns in real time as your agent runs tests or processes live data.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide Treblle with your required API Key and SDK Token.

02 Next, tell your AI client to send the full payload (request body + response body) using the `ingest_api_data` tool.

03 Finally, the data appears in your Treblle dashboard, where you can view performance metrics, check for errors, and access masked logs.

The bottom line is that your agent handles all the messy logging details; you just get clean, actionable observability data.

Built For

This MCP solves the problem of manual API logging. It's for the Backend Developer who gets frustrated having to jump between multiple dashboards just to debug a single failed endpoint. Or the DevOps Engineer tired of relying on limited terminal logs when tracing complex, multi-service failures.

Backend Developer

Uses this MCP to quickly log and test API interactions, building comprehensive documentation automatically as they write code.

DevOps Engineer

Monitors API health across different environments from the terminal, tracing traffic patterns without needing manual instrumentation scripts.

QA Engineer

Captures full context of API errors during testing phases, ensuring every failure is logged with enough detail to reproduce it later.

What Changes When You Connect

01 Stop guessing why an API failed. By using the `ingest_api_data` tool, you get full request and response payloads, allowing your agent to show you exactly where the breakdown happened.

02 Security is handled automatically. Treblle masks sensitive fields like SSNs and credit card numbers before logging anything. You gain observability without sacrificing compliance.

03 Build better documentation faster. Every time you log a successful call using this MCP, you are building out a detailed record of that endpoint's expected behavior.

04 Improve debugging cycles dramatically. Instead of relying on fragmented terminal outputs, you can trace complex error flows and performance metrics right inside Treblle.

05 Context is king. You can attach custom metadata like user IDs or environment names to every log entry, making it simple to filter thousands of calls down to the exact interaction you need.

Real-World Applications

Debugging a Payment Failure

A QA Engineer runs a payment test and gets a generic '500 Internal Server Error.' Instead of manually copying headers, they ask their agent to run the transaction and log it with `ingest_api_data`. Treblle captures the full response payload, revealing that the failure was due to an outdated card token, not a server issue.

Auditing API Changes

A Backend Developer needs to prove what data was exchanged between services last week. They use their agent to run targeted read operations and log them via `ingest_api_data`. Treblle provides a historical, auditable record of every successful call.

Troubleshooting Slow Endpoints

A DevOps team notices the `/user/details` endpoint is running slowly. They run diagnostic calls and log them to Treblle. By viewing the real-time performance metrics in the dashboard, they pinpoint that a specific database query within the response body is causing the bottleneck.

Building API Documentation

A new team member needs to understand an undocumented endpoint. They run several test calls and use `ingest_api_data` to log them. Treblle processes this stream of data, creating a clean, documented view of the request/response structure for future developers.

Patterns to Avoid

Logging only success status codes

✗ AVOID

A developer logs calls but only includes the HTTP status (e.g., '200 OK'). When an error occurs, they have no idea why it failed.

✓ INSTEAD

Use `ingest_api_data` to send the full request and response payloads. This captures *everything*, including detailed error messages within the body itself.

Ignoring sensitive data masking

✗ AVOID

Logging all raw traffic, including passwords or credit card numbers, creates a massive security risk that could lead to compliance failure.

✓ INSTEAD

This MCP handles masking automatically. The `ingest_api_data` tool ensures private fields are scrubbed before the data ever hits your dashboard.

Copy-pasting logs into spreadsheets

✗ AVOID

A QA engineer manually copies error messages and payloads from a terminal into Excel to share with the team. This process is slow, error-prone, and never captures context.

✓ INSTEAD

Let your agent use this MCP to ingest API data directly. Everything is centralized in Treblle's dashboard for instant searching and viewing.

The Right Fit

Use this MCP if you need comprehensive observability into the *structure* and *content* of your APIs—specifically, if you need to track full request/response bodies or debug performance failures. It's built for structured data logging, making it ideal for QA and backend teams.

Don't use this if all you need is a simple chat log of what the user typed into the terminal, or if you only care about high-level operational flags (e.g., 'User logged in'). For those cases, a basic message logging MCP will work better. This tool requires sending structured payloads to get value; it's not for general conversation snippets.

The Pain of Fragmented API Logging

Today, if an API call fails, you end up in a cycle of clicking. You jump from the terminal output to a separate log management system, then maybe to Jira to check ticket details. You're copying headers here, pasting payloads there, and cross-referencing timestamps across three different dashboards just to understand a single failure.

With this MCP, your agent handles all that mess. It captures the entire lifecycle of an API interaction—the request, the response, and any errors in between—and deposits it into Treblle instantly. You get one place where everything you need actually lives.

Treblle MCP: Full Visibility with `ingest_api_data`

You stop manually formatting payloads and running complex, multi-step scripts just to get a clean log. The agent handles the data structure, making sure both success and failure states are captured correctly.

The result is immediate observability. You don't have to guess what happened; you see it all logged, masked, and ready for documentation right away.

Treblle with 1 Tool

These tools allow you to send complete API request and response data directly into Treblle for instant observability.

#	TOOL	DESCRIPTION
01	`ingest_api_data`	Sends a complete set of API request and response data to Treblle, automatically masking any sensitive fields like passwords or credit card numbers.

See It in Action

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

U Ingest this API request and response data into Treblle: { "server": { ... }, "request": { ... }, "response": { ... } }

I've successfully ingested the API data into Treblle. You can now view the request details, performance metrics, and documentation in your Treblle dashboard.

U Send this API error payload to Treblle with metadata trace-id 'abc-123'.

The error payload has been sent to Treblle and tagged with trace-id 'abc-123'. It will appear in your error logs for further investigation.

U Log a successful GET request to /users into Treblle using the ingest_api_data tool.

I've logged the GET /users request to Treblle. The traffic data has been processed and is now available for monitoring.

Frequently Asked Questions

01 **How do I use Treblle MCP to document my existing APIs?**

You repeatedly run test calls through your agent using the `ingest_api_data` tool. Treblle ingests these successful interactions, automatically building out a comprehensive record of the endpoint's inputs and outputs for documentation.

02 **Does Treblle MCP mask sensitive data?**

Yes, it does. The `ingest_api_data` tool includes built-in security features that automatically detect and mask private fields like credit card numbers or passwords before they are stored.

03 Is Treblle MCP only for debugging errors?

No, it's for full observability. While excellent for error tracking, you should also use it to log successful calls to monitor performance and build documentation.

04 What data does `ingest_api_data` require?

It requires the full request payload (what was sent) and the full response payload (what came back). Sending both gives you complete context for analysis in Treblle.

05 Can I add custom labels to my API logs using Treblle MCP?

Yes. You can attach specific metadata, like unique user IDs or trace numbers, to every log entry. This allows you to filter huge amounts of data down to a very specific set of calls.

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 `"treblle": { "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

Treblle 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 Treblle. 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	Treblle MCP
Server ID	`019e38ff-04cf-7372-9c62-eff901ece6a8`
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/treblle.