MCP SERVER | NO CODE | CLOUD HOSTED

Qiita MCP

Publish and manage your technical content on Japan's dev platform.

Qiita MCP manages your entire workflow on Japan's leading developer knowledge platform. Publish articles in Markdown, search for trending tags and topics, analyze user activity, and engage with the Japanese dev community—all from your AI client.

A+ Quality Score 98.33/100

blogging | technical-writing | community | articles | qiita | japan

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

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.

Qiita MCP

29 tools available
Cloud-hosted on Vinkius

This connector gives you command over your presence on Qiita, connecting your technical writing process directly to Japan's largest developer resource. Instead of opening a browser tab, logging in, searching for tags manually, and then drafting content elsewhere, your agent handles it all. You can draft an article using Markdown right in the chat window and hit publish instantly. Need to see what topics are hot? Ask your AI client to search articles by specific tags or find out which users are making waves. It's perfect for developers targeting Japan or DevRel folks managing international content streams. When you use Vinkius, this MCP acts as a central hub, letting any compatible agent manage all your publishing, discovery, and social interactions in one spot.

Core Capabilities

01 — Publishing Content

Write new articles or update existing ones using Markdown format.

02 — Content Discovery

Search for articles, tags, and users based on keywords to track trends and research topics.

03 — Community Interaction

Like posts (LGTM), comment on discussions, or follow specific users and tags.

04 — Content Auditing

Retrieve your own published articles or check the content history of other authors.

One Click on Vinkius — From Prompt to Execution

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

01 You subscribe to this MCP and enter your specific Qiita Access Token into Vinkius.

02 Your AI client uses that token to authenticate, giving it permission to read and write content on your behalf.

03 You simply tell the agent what you want to achieve—like 'Find me three hot tags for Rust' or 'Draft an article about MCPs and publish it.' — and it executes the necessary tools.

The bottom line is, you don’t touch the Qiita website; your AI client does everything for you through this MCP.

Built For

This tool is built for technical writers and community managers who deal with high-volume, multi-platform content. If you're a DevRel needing to stay current on Japanese tech trends or an international developer building thought leadership in Asia, this MCP saves hours of context switching.

DevRel Manager

Monitors trending tags and popular users across Qiita. They use the agent to track community interest, audit content performance, and get ideas for new articles.

Technical Writer

Drafts, publishes, and updates technical guides directly from their coding environment without opening a browser or copy-pasting code snippets.

International Developer

Shares specialized knowledge with the Japanese developer community. They use this MCP to find high-traffic topics and engage in discussions via comments.

What Changes When You Connect

01 Never manually switch between Qiita and your editor again. You can draft and publish articles directly through your agent, keeping your workflow contained in one place.

02 Instead of relying on keyword searches, you can use the `search_articles` tool to filter results by advanced syntax like 'tag:python' or 'created:>2024-01-01', giving you precise data for content planning.

03 You gain deep social insights. Use tools like `get_user_followers` and `get_user_stocks` to analyze who your audience is, which users resonate, and what content they bookmark.

04 Stay current with trends effortlessly. By using `get_tags` and `get_tag_articles`, you instantly know the most popular topics and can generate content around high-interest areas.

05 Manage your reputation. You don't just publish; you engage. Use `post_comment` or `like_article` to participate in discussions, building credibility with the Japanese tech community.

Real-World Applications

Auditing content performance

A DevRel needs to know which of their last five articles are performing best. Instead of checking metrics one by one on the dashboard, they ask their agent to use `get_my_articles` and then analyze the results, immediately identifying a gap in coverage that requires an update.

Competitive content research

A developer wants to see what topics competitors are writing about. They use `get_user_articles` on a known competitor's profile and then run `search_articles` using the competitor's style, finding valuable gaps in market coverage.

Responding to community feedback

An author publishes an article and receives many comments. Instead of scrolling through pages of text, they ask their agent to run `get_comments` to get a summary and categorize the discussion points for quick follow-up.

Developing niche content streams

A tech writer needs articles on 'web assembly' but doesn't know which tags are popular. They first use `get_tags` to see the most followed topics, then narrow it down with `get_tag_articles` to find existing high-quality examples.

Patterns to Avoid

Copying content manually

✗ AVOID

A user reads three articles on a topic and tries to copy the best ideas into a new draft, forcing them to switch between browser tabs.

✓ INSTEAD

Use `get_article` or `search_articles` with specific filters. Your agent gathers the necessary raw data (text, tags) for you in one go.

Forgetting revisions

✗ AVOID

An article is published but has a typo or needs updated tag information months later.

✓ INSTEAD

Don't recreate it; use `get_my_articles` to find the ID and then run `update_article`. This ensures you keep the original comment count and history.

Blind engagement

✗ AVOID

Reading a bunch of articles but having no way to track which ones are truly useful or important for later reference.

✓ INSTEAD

Before reading, use `search_articles` to find content. After finding something good, hit the agent and tell it to `stock_article` so you can bookmark it without leaving your workflow.

The Right Fit

Use this MCP if your primary goal is technical knowledge sharing within the Japanese developer ecosystem—whether that means publishing guides or tracking trends. This tool handles everything from basic content creation (`publish_article`) to deep audience analysis (`get_user_followers`). Don't use it if you just need to read random articles; for that, a simple web browser works fine. Crucially, don't try to manage your entire portfolio through this MCP if you are only concerned with one type of content (like just comments). In that case, focus on the `post_comment` tool and ignore everything else. This MCP is designed for holistic account management, covering publishing, discovery, and social interaction.

The Friction of Cross-Platform Content Management

Right now, if you want to publish an article on Qiita, you open the site. You write in your local editor, then copy/paste into a draft box. If you need to check what tags are popular or see what competitors wrote last week, you have to run multiple searches and manually cross-reference dates and topics across different browser tabs.

With this MCP, all that friction disappears. Your agent treats the entire Qiita platform like an internal database. You just ask it to find trending topics using `get_tags`, then draft the content locally, and finally tell your AI client to publish it with one command. It's instant.

Get Full Control Over Your Qiita Presence

You no longer have to manually remember which articles you need to update or who followed whom last week. The agent runs `get_my_articles` for an audit, and then if a section is weak, it uses the data from `get_tag_detail` to suggest better tags.

What's different now is control. You dictate the content strategy—whether you need to follow specific users with `follow_user`, or simply track overall market interest using `search_articles`. The tool adapts to your goals, not the other way around.

Qiita: 28 Tools for Developers

These tools let you perform every action on Qiita—from writing full articles to analyzing user connections—all through conversation with your AI client.

#	TOOL	DESCRIPTION
01	delete_article	Permanently removes an article, along with all associated comments, likes, and bookmarks.
02	delete_comment	Removes a comment from Qiita; you can only delete comments that you personally wrote.
03	edit_comment	Updates the text of an existing comment on a Qiita article, but only if you authored it.
04	follow_tag	Adds a tag to your personalized feed so articles using that topic show up regularly.
05	follow_user	Subscribes to content from a specific user, ensuring their new posts appear in your feed.
06	get_article	Retrieves the complete text and structure of a single Qiita article using its ID for analysis or review.
07	get_comments	Fetches all comments left on an article, allowing you to analyze community feedback quickly.
08	get_my_articles	Retrieves a list of articles that you have published yourself for auditing or updating.
09	get_my_profile	Provides your authenticated user details and overall community standing on Qiita.
10	unfollow_tag	Removes a tag from your personalized feed, stopping notifications about that topic.
11	get_tag_articles	Finds all articles associated with a specific tag for content research or trend spotting.
12	get_tag_detail	Shows detailed information about a specific Qiita tag, helping you assess its popularity before writing.
13	get_tags	Lists popular tags on the platform, sorted by follower count to guide your content strategy.

#	TOOL	DESCRIPTION
14	get_teams	Retrieves a list of corporate or organizational teams you are a member of on Qiita.
15	get_user_articles	Gets all articles published by another specific user, useful for competitive research.
16	get_user_followees	Lists the users that a given account is following, helping map out influence networks.
17	get_user_followers	Retrieves the list of people who follow a specific user to understand your audience size.
18	stock_article	Adds an article to your private collection, similar to bookmarking it for later reading.
19	get_user_stocks	Retrieves articles that another user has bookmarked, giving insight into their interests.
20	get_user	Pulls the profile information of any Qiita user using just their username for identity verification.
21	is_item_stocked	Checks if a specific article is already in your bookmarks before you try to add it.
22	like_article	Registers an appreciation (LGTM) on an article, boosting its visibility and signaling value to the author.
23	post_comment	Allows you to write a comment in response to an article or discussion thread.
24	publish_article	Writes and publishes a brand-new article using Markdown format, letting you set up to five tags.
25	search_articles	Searches the entire Qiita database using advanced syntax filters like 'tag:python' or 'user:username'.
26	unfollow_user	Stops receiving updates from a user, keeping your feed focused on relevant content.
27	unlike_article	Removes your appreciation (LGTM) from an article, decreasing its public count by one.
28	unstock_article	Removes a specific article from your personal collection of bookmarks.
29	update_article	Modifies the title, body content, tags, or visibility status of an existing published piece.

See It in Action

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

U Search for the latest articles on Qiita about 'MCP' and 'AI agents'.

I've found several recent articles on Qiita about MCP and AI agents. The top ones include discussions about building MCP servers in TypeScript and integrating AI agents into development workflows. Would you like to see details for any of these?

U Create a new article on Qiita titled 'Getting Started with MCP Servers' with a brief introduction.

I've published the article 'Getting Started with MCP Servers' on your Qiita account. It's now live and accessible to the community.

U Show me my published articles on Qiita.

Here are your published articles on Qiita. You have 5 articles, with the most recent being 'Advanced TypeScript Patterns'. Would you like to update or manage any of them?

Frequently Asked Questions

01 **How do I find trending topics using Qiita MCP?**

You use the `get_tags` tool. This function pulls a list of popular tags on Qiita, sorted by follower count so you know exactly where to focus your next article.

02 **Can I update an article using the Qiita MCP?**

Yes, use `update_article`. This tool lets you modify any aspect of a published piece—like fixing typos or changing tags—without losing its original history or comments.

03 **Does the Qiita MCP help me analyze my audience?**

Absolutely. You can use `get_user_followers` and `get_user_stocks` to track who follows you, helping you understand your audience size and their specific interests.

04 **What if I want to delete content on Qiita?**

You have dedicated tools for this. Use `delete_article` or `delete_comment`, but remember that both actions are permanent, so double-check your intent before running them.

05 **Is the Qiita MCP only for English content?**

No. It connects to the entire platform's API, allowing you to publish and manage content written in any language that fits the Markdown format.

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 `"qiita": { "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

Qiita 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 Qiita. 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	Qiita MCP
Server ID	`019ef45e-3506-703c-948b-79e3bdb38656`
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/qiita.