MCP SERVER | NO CODE | CLOUD HOSTED

Tumblr MCP

Manage Community Content & Trends

Tumblr MCP connects your AI agent directly to Tumblr's content infrastructure. Use it to analyze a blog's performance, find trending topics by searching specific tags, and list post histories for any account. It lets you manage complex social monitoring tasks—from checking an avatar URL to pulling the full details on individual posts—all through natural conversation.

A+ Quality Score 100/100

blogging | content-discovery | social-networking | multimedia | publishing | community

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

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.

Tumblr MCP

5 tools available
Cloud-hosted on Vinkius

Managing a presence on Tumblr used to mean jumping between analytics dashboards and content feeds just to gather enough data for a strategy meeting. Now, your AI agent handles that complexity. You can ask it to find out everything about a specific blog's history or pull all the latest posts matching a certain theme. Need to know what's popular right now? Just tell your agent to search across tags; it pulls those trends immediately. The whole process is managed through natural conversation, giving you deep oversight of your microblogging activity without ever opening the official site interface. Because this MCP sits in the Vinkius catalog, you connect once and get access to these powerful blogging tools alongside hundreds of others.

Core Capabilities

01 — Check blog identity details

Retrieve basic information, like the name or follower count, for any Tumblr account.

02 — Find a blog's profile picture

Get the URL needed to display an avatar image for a specific blog.

03 — Search posts by topic tag

List all recent content across Tumblr that shares a particular hashtag or keyword.

04 — Pull content from a specific account

Generate a list of published articles from one designated blog, allowing you to filter by content type.

05 — Get deep details on an article

Fetch complete metrics and all available data for one single post.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your Tumblr API Consumer Key in the Vinkius Marketplace.

02 Connect your preferred AI agent (like Cursor or Claude) to the Vinkius platform.

03 Ask your agent a question like, 'List all photo posts from the blog X,' and it handles the data retrieval.

The bottom line is you talk naturally to your agent, and it uses these specific tools to pull structured Tumblr data for you.

Built For

This MCP is built for anyone who treats their online presence as a measurable asset. It's for the social media manager who needs instant trend reports, the content creator who struggles to find unique angles, or the data analyst tracking community engagement over time.

Social Media Manager

Monitors multiple accounts to track trending tags and pull post histories for weekly performance reports.

Content Creator

Uses the agent to analyze a blog's metadata or retrieve full details on specific posts to plan better content angles.

Digital Marketing Analyst

Systematically lists and compares post data from several different blogs across the platform to measure competitive performance.

What Changes When You Connect

01 Instantly discover what's hot by using `list_tagged_posts`. Instead of manually browsing tag feeds, your agent searches the entire platform and gives you a list of posts matching any topic.

02 Conduct deep competitive audits. You can use `get_blog_info` to get key statistics on a rival blog, or use `list_blog_posts` to track their content output over time.

03 Save hours analyzing individual pieces. Instead of manually clicking into every post to check its stats, you just ask your agent and it uses `get_post` to pull the complete data set for you.

04 Streamline cross-platform reporting. You can quickly get a list of posts from multiple accounts using `list_blog_posts`, ensuring all your content sources are accounted for in one query.

05 Build structured knowledge bases. By combining `get_blog_avatar` and `get_blog_info`, you build out a clean index of key community players for research purposes.

Real-World Applications

Identifying new content angles

A brand manager needs to know what topics are gaining traction. They ask their agent, 'Show me the latest posts tagged with sustainable fashion.' The agent uses `list_tagged_posts` and returns a curated feed of ideas, giving them instant inspiration.

Auditing an account's content history

A marketing team needs to review all photo blog posts from the 'official brand' account. They instruct their agent to use `list_blog_posts` and filter by 'photo', getting a clean list of everything published in that format.

Quick competitor snapshot

A user wants to know how big a rival blog is before starting. They ask the agent for basic information, and it uses `get_blog_info` to immediately provide key stats like follower count and general metadata.

Deep-dive post analysis

A user finds one specific post they suspect is performing well. They ask the agent to pull all data on that ID, triggering `get_post`, which returns engagement metrics, content details, and more for immediate analysis.

Patterns to Avoid

Asking for a general search

✗ AVOID

Saying 'Give me all posts about travel.' This is too broad because the agent needs specific tags or blog names to function.

✓ INSTEAD

Use `list_tagged_posts` and specify the exact tag, like 'Show me posts tagged with #travelphotography.' Or, use `list_blog_posts` if you want content from a known account.

Assuming all data is available

✗ AVOID

Expecting to find every single piece of historical data in one go. The platform structures data into specific views.

✓ INSTEAD

To get comprehensive details, always ask for the full metrics by using `get_post` with a specific post ID.

Mixing up blog info and posts

✗ AVOID

Asking 'What's the best content on this site?' The agent can’t guess; it needs to know if you want general account stats or an actual list of articles.

✓ INSTEAD

If you need account details, use `get_blog_info`. If you need a list of published pieces, use `list_blog_posts` and specify the blog name.

The Right Fit

Use this MCP if your workflow requires structured access to Tumblr's specific data layer. You need to analyze metrics, compare multiple accounts, or track content performance using tools like `get_post`, `list_tagged_posts`, and `list_blog_posts`. Don't use it if you are just trying to write a blog post from scratch; that’s pure generation work. Also, don't rely on this MCP for real-time live interaction—it fetches data based on the current API snapshot. If your goal is simply general web browsing or accessing unstructured text that doesn't relate to a specific blog account or tag, you probably need a different type of agent connector instead.

Tracking content performance used to be a tedious process of clicking through tabs and exporting spreadsheets.

Before this MCP, getting an overview required logging into Tumblr, navigating to the analytics dashboard, then separately going through tag feeds to see what was trending. You'd spend fifteen minutes just copying post IDs and pasting them into a spreadsheet—a messy cycle of clicks and context switching.

Now, you simply ask your agent for the information. Whether you need the basic details on an account using `get_blog_info` or want to track every single article from a specific source via `list_blog_posts`, you get the clean, structured data dump immediately.

The Tumblr MCP gives you instant visibility into content performance.

You no longer have to manually search for posts. Instead of clicking through tag by tag, your agent executes `list_tagged_posts` and aggregates the results instantly, showing you exactly what's hot across the entire platform at once.

The biggest difference is that data gathering isn't a series of manual steps anymore; it’s one conversation. You get immediate, comprehensive answers instead of hours of clicking through dashboards.

Tumblr MCP: 5 Tools for Blogging Analysis

Use these tools to retrieve everything from single post details to broad platform trends across Tumblr's entire content catalog.

#	TOOL	DESCRIPTION
01	`get_blog_avatar`	Retrieves only the URL needed for a blog's profile picture.
02	`get_blog_info`	Gathers general data about any given Tumblr blog, including its name and status.
03	`get_post`	Pulls all available metrics and details for one specific post ID.
04	`list_blog_posts`	Generates a list of articles from a known blog, letting you filter by content type like photo or text.
05	`list_tagged_posts`	Searches the entire platform and lists all recent posts that share a specific hashtag.

See It in Action

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

U Show me the latest posts tagged with 'illustration'.

I've retrieved the latest tagged posts. Here are some beautiful illustrations trending right now from creators across Tumblr. Would you like the details for any specific post?

U List all photo posts from the blog 'officialtumblr'.

I've fetched the photo posts from 'officialtumblr'. You have 5 recent images including anniversary graphics and community features. Shall I retrieve the full description for the latest one?

U Get information about the blog 'staff'.

Fetching blog info... The 'staff' blog is the official Tumblr internal blog. It has over 1M followers and focuses on product updates and platform news. Would you like to see their latest posts?

Frequently Asked Questions

01 **How do I connect the Tumblr MCP?**

You first subscribe to this MCP on Vinkius and must provide your specific Tumblr API Consumer Key. Once that key is saved, your AI agent can access all the tools.

02 **Can the Tumblr MCP find posts from a non-existent blog?**

No. The agent requires an active hostname or account name to query data using `get_blog_info` or `list_blog_posts`. It won't guess.

03 **What if I need data for a specific post? Which tool do I use?**

You must use the `get_post` tool. This requires you to provide the unique ID of the article, which gives you full access to its engagement metrics.

04 **Is this MCP only for photos and text posts?**

No. While you can list content types using `list_blog_posts`, the agent pulls general metadata that covers all multimedia formats, including quotes and videos.

05 **Can I use Tumblr MCP to find out about a competitor's account?**

Yes. You can get an initial overview of their profile using `get_blog_info` or check their recent posts with `list_blog_posts`.

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 `"tumblr": { "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

Tumblr 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 Tumblr. 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	Tumblr MCP
Server ID	`019dd17a-69b6-70d3-bc31-8908e6f202e4`
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/tumblr.