MCP SERVER | NO CODE | CLOUD HOSTED

Freshchat MCP

Manage Chats and Customer Data Via AI Commands

Freshchat connects your agent workflows directly into your existing chat platform through Model Context Protocol (MCP). Use this tool to manage customer conversations across web, mobile, and social channels. Your agent can track active chats, fetch detailed user profiles by email, retrieve full message histories for auditing, and even post real-time updates without you leaving your primary workspace.

A+ Quality Score 100/100

live-chat | messaging | customer-profiles | conversation-management | support-automation

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

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.

Freshchat MCP

12 tools available
Cloud-hosted on Vinkius

This MCP lets your AI client interact with Freshchat, giving it the ability to handle customer messaging and support workflows automatically. It's built for sales teams and dedicated support staff who need a single source of truth about every interaction.

You can use your agent to see which chats are currently open or resolved, get deep details on any chat participant's profile, and look up full message logs instantly. You don't have to manually jump between tabs anymore. Need to follow up? Your agent can send a new message directly into an ongoing conversation. If you're building custom workflows, connecting through the Vinkius Marketplace makes it simple to bring all this data—from user records to active chat status—into any compatible AI client for immediate action.

Core Capabilities

01 — Monitor Chat Status and Details

List every open or resolved conversation, and retrieve detailed metadata about the conversations themselves.

02 — Locate User Information

Find customer profiles using an email address or check the status of agents on the support team.

03 — Review Communication History

Get a full transcript and history of messages for any specific conversation ID, ensuring you have complete audit trails.

04 — Send Real-Time Updates

Post new follow-up messages directly into an existing customer chat thread.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on the Vinkius Marketplace and provide your Freshchat Region Domain and API Token.

02 Connect your AI client (like Cursor or Claude) to the configured MCP endpoint.

03 Instruct your agent with natural language commands; it will then execute tools like list_conversations or search_chat_users to get the data you need.

The bottom line is, once connected, your AI client handles all the complex API calls so you can simply talk to your chat system.

Built For

This MCP is built for support operations leads and customer success managers who spend too much time context-switching. It's for anyone whose job requires knowing the status of a conversation, finding a specific user, or summarizing chat history without clicking through five different dashboards.

Support Lead

Needs to check agent availability using list_support_agents and quickly view all open chats via list_conversations without leaving their primary dashboard.

Customer Success Manager

Uses the MCP to find a customer's full profile with get_chat_user_details, then uses search_chat_users to verify identity before sending a follow-up message.

Sales Engineer

Needs to audit past interactions by calling list_chat_messages and verifying chat history for lead qualification purposes.

What Changes When You Connect

01 Instantly review conversations: Instead of opening the main dashboard to see open chats, your agent can call list_conversations to give you a real-time summary. This saves minutes on every single shift.

02 Contextual user data: You don't need to guess who the person is. By calling get_chat_user_details, your agent pulls up full profile information right into your chat flow, giving immediate context for problem solving.

03 Deep audit trail access: Need to prove when a message was sent? Use list_chat_messages to pull the complete history of any conversation ID. This is non-negotiable for compliance and reporting.

04 Proactive messaging: Don't wait for the customer to reply. With send_chat_message, you can automatically post follow-up messages or status updates right into the chat thread using simple prompts.

05 Team visibility: Get an immediate overview of who is available by listing support agents with list_support_agents. You know exactly who to assign a ticket to without checking multiple statuses.

Real-World Applications

Handling VIP Customer Escalations

A CSM needs context on a high-value client. They ask their agent to find the user by email using search_chat_users. The agent confirms the ID and then uses get_chat_user_details to confirm their purchase history before escalating the issue.

Closing Out an Old Ticket

A support lead is reviewing old tickets for billing issues. They ask the agent to list_conversations, identify a closed chat, and then use list_chat_messages on that conversation ID to pull out the full transcript needed for the final report.

Post-Sale Follow Up

A sales engineer needs to follow up after a demo. They ask their agent to send_chat_message with an update, ensuring the customer sees the reply immediately in the active chat widget without manual intervention.

Agent Handoff Verification

When handing off a ticket, the team needs background. They use get_conversation_details to pull all metadata and list_chat_users to confirm which department was involved, ensuring no critical details were missed.

Patterns to Avoid

Assuming chat status is available

✗ AVOID

The user just sees a blinking cursor in the web interface and assumes it's open. They waste time trying to find the right button or person.

✓ INSTEAD

To confirm if a conversation is truly active, prompt your agent to run list_conversations. This gives you structured data showing all statuses (open/resolved) immediately.

Asking for user details by name

✗ AVOID

The user knows the customer's name but can't find them in the chat system because they only have a generic directory login.

✓ INSTEAD

Always use search_chat_users and provide the email address. This precise input guarantees you get the correct Freshchat user record.

Missing context on old threads

✗ AVOID

A support agent jumps into a 6-month-old thread, but only sees the last message, not the reason for the issue or previous attempts to fix it.

✓ INSTEAD

To get the full picture, prompt your agent to list_chat_messages for that specific conversation ID. This retrieves the entire historical log.

The Right Fit

Use this MCP if your primary pain point is context switching across different customer service dashboards—specifically when you need to check chat status, pull user data by email, or append messages to an ongoing thread. If you are doing that, this is the right tool.

Don't use this if your main goal is managing internal knowledge bases or connecting to HR records; for that, you need a different category of MCP. Also, don't rely on it just to see *all* users in your

company directory—for general user lists outside of chat context, check out the generic Identity Management tools instead.

Sifting through support conversations feels like digging through digital archives.

Right now, if a customer calls about an issue that started last week, you have to manually open the chat system, find their name or email, click into the conversation thread, scroll up past dozens of messages, and piece together what happened. Then you might have to switch tabs entirely just to verify who they are in your main CRM.

With this MCP connected through Vinkius, the process changes completely. Instead of clicking, you tell your agent: 'Show me everything about user X.' The agent runs search_chat_users and immediately pulls up get_chat_user_details alongside all relevant chat history via list_chat_messages. You get a single, actionable summary.

Getting Full Conversation Context with Freshchat MCP

The tedious steps that disappear are the manual searches for user IDs and the labor of scrolling through message threads to find the root cause. You no longer need to copy a conversation ID from one window and paste it into another tool just to get metadata.

What's different now is speed and completeness. Your agent gives you all the required data—the full list_chat_messages, the user profile details, and the current status of list_conversations—in one continuous flow, letting you resolve issues faster than ever.

Freshchat: 12 Tools for Messaging Management

These tools allow your AI agent to perform specific actions within Freshchat. They let you list agents, find users by email, review message history, and send real-time updates.

#	TOOL	DESCRIPTION
01	`check_account_status`	Verifies the current configuration status of your Freshchat account.
02	`get_agent_profile`	Retrieves detailed metadata and information about a specific support agent.
03	`get_conversation_details`	Fetches comprehensive metadata describing an entire chat conversation.
04	`get_chat_user_details`	Retrieves detailed profile information for any participant in a chat.
05	`list_support_agents`	Provides a list of all available support agents and team members within the organization.
06	`list_chat_channels`	Lists all configured messaging channels that your team uses for routing conversations.
07	`list_conversations`	Retrieves a list of currently active chats and their general metadata.
08	`list_agent_groups`	Lists the different agent groups or teams defined in your Freshchat setup.
09	`list_chat_messages`	Fetches a chronological list of all messages exchanged within a specific chat conversation ID.
10	`list_chat_users`	Lists all individual participants who have been involved in a given chat thread.
11	`search_chat_users`	Searches the user database to find a specific Freshchat user using their email address.
12	`send_chat_message`	Posts a new message into an existing chat conversation thread for immediate customer visibility.

See It in Action

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

U List all open conversations in my Freshchat account.

Retrieving chats... I found 4 open conversations, including one with 'John Doe' regarding 'Billing' and another with 'Sarah' about 'API Integration'.

U Find the Freshchat user with the email 'customer@example.com'.

Searching users... I found a record for 'Alice Smith' (ID: user_abc123) associated with that email. She has 2 past conversations in your account.

U Send a message to conversation 'conv_987': 'I am looking into this for you'.

Message sent! Your update has been successfully posted to conversation 'conv_987'. The customer will see your reply in their chat widget.

Frequently Asked Questions

01 **How do I check if a chat conversation is active using Freshchat MCP?**

You use the list_conversations tool. This sends a list of all chats and their status metadata, letting you quickly filter for those that are currently open or resolved.

02 **Can I find a user's profile if I don't know their chat ID?**

Yes, use search_chat_users. You only need the customer's email address to run this tool and retrieve a dedicated record of their details.

03 **What is the difference between list_conversations and get_conversation_details?**

list_conversations gives you a summary view—a quick list of all chats. get_conversation_details pulls deep, specific metadata on one single chat conversation ID.

04 **How do I make sure the customer sees my follow-up message?**

Use send_chat_message. This tool posts a new update directly into the existing conversation thread, ensuring the customer receives your reply in their active chat widget.

05 **Can Freshchat MCP list all people who have been in a chat?**

Yes, running list_chat_users on a specific conversation ID provides a clear roster of every unique participant involved in that thread.

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 `"freshchat": { "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

Freshchat 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 Freshchat. 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	Freshchat MCP
Server ID	`019d75a0-4a0f-7267-a702-316527c19177`
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/freshchat.