MCP SERVER | NO CODE | CLOUD HOSTED

Sendbird MCP

Manage chat users and moderation instantly.

Sendbird gives your AI agent total command over a complex chat ecosystem. Use this MCP to manage users, create everything from massive public open channels to private group chats, and enforce moderation rules instantly. You can control user accounts, run automated bots, and ban or mute troublesome members—all through natural conversation.

A+ Quality Score 100/100

in-app-chat | messaging-api | user-management | real-time-communication | moderation

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

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.

Sendbird MCP

18 tools available
Cloud-hosted on Vinkius

You connect your messaging infrastructure directly to any AI agent via Vinkius. Sendbird lets you handle every aspect of your chat platform without logging into a separate dashboard. Instead of navigating complex menus to check who's online or banning a spammer, your AI acts as a real-time administrator.

This MCP gives you the ability to build out and maintain an entire communication ecosystem through simple commands. You can create new users and manage their profiles, set up both public open channels for general discussion, and private group chats for specific teams. If you need to enforce safety rules, your agent handles moderation actions like blocking or muting troublesome members. Plus, you can programmatically deploy bots that send messages and interact with users on your behalf.

Core Capabilities

01 — Manage User Accounts

Create new user profiles, list existing accounts, or modify user tokens within the chat system.

02 — Control Channel Creation and Lifecycle

Establish both large public channels and private group chats, and manage them by updating metadata or deleting them entirely.

03 — Enforce Safety Rules

Maintain community standards by muting users in a channel, banning accounts outright, or freezing entire conversations during an incident.

04 — Automate Messaging and Bots

Set up and manage bots to send automated messages and programmatically interact with different user groups.

05 — Navigate Channel Status

Retrieve information about specific open channels or list all existing public channels for quick reference.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to the Sendbird MCP and provide your required Sendbird App ID and Master API Token.

02 Connect this MCP to any compatible AI client, granting it permissions to manage your chat environment.

03 Ask your agent conversational questions like 'Ban user X for violating community guidelines,' or 'Create a new group channel for the marketing team.'

The bottom line is you use natural conversation to execute administrative tasks that used to require multiple clicks and complex dashboard navigation.

Built For

This MCP is essential for Support Leads who are tired of switching between ticketing systems and chat dashboards. It's also perfect for Developers testing user provisioning or Community Managers who need instant, large-scale moderation tools without leaving their console.

Support Lead

Needs to quickly check channel history and moderate users (like using 'mute_user' or 'block_user') without exiting the support agent interface.

Community Manager

Oversees large, public channels. They use this MCP to enforce safety rules across huge user bases by calling tools like 'ban_user' and 'freeze_channel'.

Software Developer

Tests new features by writing scripts that call tools such as 'create_open_channel' or 'create_group_channel' to provision environments.

What Changes When You Connect

01 Moderation is immediate. Instead of checking a dashboard to ban someone, you just ask your agent to 'ban user X,' and it happens instantly across the entire platform.

02 Channel setup scales with natural language. You don't need to manually create an open channel or group; you simply tell your AI client to do it using 'create_open_channel.'

03 You maintain full control over user accounts. Use 'list_users' to see every account and 'block_user' if someone needs immediate removal from the system.

04 Automation is simple. You can create dedicated bots, then use 'send_bot_message' to distribute automated announcements or support responses without writing any boilerplate code.

05 Safety protocols are centralized. If a public channel gets into trouble, you don't waste time; you run 'freeze_channel' to stop the mess instantly.

Real-World Applications

Handling Spam in a Public Forum

A community manager notices a spammer posting repeatedly in the main forum. They ask their agent: 'Freeze channel at URL X and ban user Y.' The MCP executes both commands immediately, stopping the problem without manual intervention.

Onboarding a New Department

A developer needs to set up private communications for three new teams. They ask their agent: 'Create group channels and invite users A, B, C.' The MCP uses 'create_group_channel' and 'invite_group_channel' multiple times, provisioning the entire communication structure in seconds.

Managing Support Escalations

A support lead needs to temporarily silence a user who is harassing others. They simply tell their agent: 'Mute this user.' The MCP runs 'mute_user,' solving the issue instantly without needing to access moderation tools.

Auditing User Activity

A compliance officer needs an audit of all platform users. They ask their agent to run 'list_users' and receive a comprehensive list, allowing them to verify account status across the entire system.

Patterns to Avoid

Trying to manually update channel details

✗ AVOID

The user tries to go into the dashboard settings, find the correct open channel, and manually change its name. This takes multiple clicks and is prone to human error.

✓ INSTEAD

Use the 'update_open_channel' tool via your agent. You just need to tell it, 'Change the metadata for this open channel to X,' which bypasses all manual clicking.

Forgetting user permissions

✗ AVOID

The support team tries to block a high-profile account but doesn't know if they have the necessary API token permissions, leading to an error.

✓ INSTEAD

Ensure your setup includes all tokens and use 'block_user' through your agent. The MCP handles the permission checks and execution flow for you.

Using a basic messaging tool

✗ AVOID

The user relies only on sending messages, but cannot perform administrative actions like freezing or banning users.

✓ INSTEAD

This MCP provides deep control. Use 'freeze_channel' to halt all communication immediately, which is far beyond simple message sending.

The Right Fit

You should use this Sendbird MCP if your primary need is administrative control over a live chat platform—specifically managing users, enforcing moderation, and building channels at scale. It's perfect for teams that need to move from manual dashboard clicks to conversational command execution.

Do NOT use this MCP if you only need simple messaging functionality (e.g., just sending one-off messages). For those basic tasks, a simpler integration might suffice. However, if your workflow involves any level of moderation—like using 'ban_user' or managing complex user lifecycles with 'create_user'—this MCP is required because it gives you the full administrative toolkit.

Handling chat infrastructure used to mean a dozen different dashboards and tabs.

Today, if your team needs to moderate an issue, you typically have to jump between three systems: the user management portal, the channel settings page, and the moderation log. You copy IDs here, paste them there, click through multiple confirmation screens, and then repeat the process for every single incident.

With this MCP, that entire manual dance disappears. Your agent handles it all in one go. Instead of copying IDs or clicking a dozen times, you simply ask your AI client to perform the moderation actions—like 'mute_user' or 'block_user.' You get direct control without leaving your primary workflow.

Sendbird MCP gives you full command over user and channel lifecycle management.

The manual steps that vanish include listing users across different tabs, creating a new public space (using 'create_open_channel'), and then finding the right group to invite people into. It's a multi-step process with high friction every single time.

Now, you tell your agent what needs doing—for instance, 'Create a global channel and populate it with these five users.' The MCP runs all necessary tools sequentially, handling the entire workflow from start to finish in one command.

Sendbird: 17 Tools for Chat Management

These tools let you programmatically handle every administrative function of your chat system, from creating accounts to enforcing safety rules.

#	TOOL	DESCRIPTION
01	ban_user	Removes a user from a channel permanently using the ban function.
02	block_user	Prevents a specific user from interacting with your chat application.
03	create_bot	Registers and sets up a new automated bot account within the messaging system.
04	create_group_channel	Create a group channel
05	create_open_channel	Establishes a public channel accessible to all users in your community.
06	create_user	Create a new Sendbird user
07	delete_open_channel	Permanently removes an open channel from the system.
08	freeze_channel	Stops all messaging in a channel, preventing users from posting until it is unfrozen.
09	get_open_channel	Retrieves the details and status of a specific open channel using its URL.
10	invite_group_channel	Adds existing users to a private group chat that they are not already part of.
11	join_group_channel	Makes the calling user join an existing private group conversation.
12	leave_group_channel	Removes the calling user from a specific private group chat.
13	list_open_channels	Generates a list of all current public channels available on the platform.
14	list_users	Retrieves a comprehensive list of every user currently registered on the platform.
15	mute_user	Silences a user within a channel, preventing them from sending messages without full removal.
16	send_bot_message	Sends an automated message through the established bot account to a target user or group.
17	send_message	Allows sending any standard text message into a specified channel.

#	TOOL	DESCRIPTION
18	update_open_channel	Modifies the metadata or settings of an existing public channel, like changing its name.

See It in Action

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

U List the first 10 users in our Sendbird application.

I've retrieved the user list. Found users: 'admin_01', 'support_agent_a', 'user_992'... Would you like to see the profile details for any of them?

U Create a new open channel called 'Global-Lounge' for our community.

Channel 'Global-Lounge' has been created successfully. The channel URL is 'sendbird_open_channel_552'. You can now start directing users to this public space.

U Freeze the channel at URL 'sendbird_open_channel_123' to stop all messaging.

The channel 'sendbird_open_channel_123' is now frozen. Users will be unable to send new messages until it is unfrozen.

Frequently Asked Questions

01 **Can I use Sendbird MCP to list all existing users?**

Yes. You can run 'list_users' through your agent to retrieve a comprehensive roster of every user on your platform, which is helpful for auditing or segmentation.

02 **How do I create a new public channel using Sendbird MCP?**

You use the 'create_open_channel' tool. This allows you to establish a massive public space and get its URL immediately, letting your community direct traffic there.

03 **What is the difference between ’block_user’ and ’ban_user’ in Sendbird MCP?**

While both restrict access, 'block_user' generally prevents interaction with the application. 'ban_user,' however, performs a more permanent ban from the specific channel.

04 **Can I automate announcements using Sendbird MCP?**

Yes. You can use the 'create_bot' tool to set up an automated account and then execute 'send_bot_message' to distribute announcements across your user base programmatically.

05 **What if I need to delete a public channel?**

You run the 'delete_open_channel' tool via your agent. This permanently removes the specified open channel from the system, cleaning up unnecessary spaces.

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 `"sendbird": { "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

Sendbird 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 Sendbird. 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	Sendbird MCP
Server ID	`019e38ea-9c83-70a6-9848-f5cb135bc104`
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/sendbird.