MCP SERVER | NO CODE | CLOUD HOSTED

Frontegg MCP

Audit and manage B2B identities in plain conversation

Frontegg connects your agent to a B2B SaaS identity management platform, letting you programmatically handle user accounts and multi-tenant structures. Use this MCP to list all customer accounts, audit roles and permissions, or onboard new users instantly from any compatible AI client. It's built for DevOps teams that need real-time visibility into who has access to what across complex environments.

A+ Quality Score 100/100

b2b-saas | user-management | authentication | multi-tenancy | provisioning | audit-logs

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

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.

Frontegg MCP

12 tools available
Cloud-hosted on Vinkius

Managing B2B identities gets complicated fast. When you have multiple customers (tenants), keeping track of user roles and permissions manually is a nightmare. This MCP connects your agent directly to the core logic of the Frontegg platform, letting you manage all that identity data without ever touching an admin dashboard. You can list every tenant account and instantly pull detailed profiles for any global user. Need to onboard someone or shut down access? Your agent handles it in plain conversation. It's exactly what you need when you want your AI client to treat your B2B SaaS environment like a database, rather than a confusing web UI. By connecting this MCP through Vinkius, you get instant command-line control over complex user and tenant lifecycles.

Core Capabilities

01 — Manage Tenant Lifecycle

List all current customer accounts, retrieve their specific configuration details, or programmatically create and delete entire tenants.

02 — Control User Accounts

Access the global user database to fetch detailed profiles for any account, and instantly provision or remove individual users.

03 — Audit Access Control Models

List all available system roles and granular permissions across the entire platform, allowing you to audit security settings efficiently.

04 — Retrieve Integration Keys

Fetch Machine-to-Machine tokens for specific tenants when setting up back-end integrations.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP through the Vinkius Marketplace.

02 Enter your Frontegg Client ID and API Key, which you find in your vendor workspace.

03 Start giving commands to your agent from Claude, Cursor, or any compatible client.

The bottom line is that you talk to your AI client naturally, and it executes complex identity management actions directly against the Frontegg system.

Built For

This MCP is for operational teams who deal with complex, multi-client SaaS architecture. If you're an IT Admin tired of navigating dozens of dashboards just to check a user's role or provision a new client, this is for you.

DevOps Engineer

Needs to quickly verify tenant configurations or adjust user access levels across hundreds of clients without manual dashboard navigation.

Product Manager (PM)

Wants a real-time, auditable overview of customer provisioning status and role assignments for new features via simple AI commands.

Support Engineer

Must automate the retrieval of user IDs and tenant associations when troubleshooting access issues or handling support tickets.

What Changes When You Connect

01 You can check the status of your API connection using `check_environment_status` before running any commands, ensuring zero downtime when provisioning users or managing tenants.

02 Instead of clicking through tenant dashboards to see who belongs where, you can use `list_tenants` and then `get_user_details` to instantly cross-reference a user's full profile and assigned account ID.

03 Auditing security models is simple: run `list_system_roles` and `list_permissions` to get a complete picture of every access level available across your entire SaaS setup.

04 You gain control over the entire customer lifecycle by executing commands like `create_tenant` or `delete_user`, letting your agent handle the heavy lifting in one prompt.

05 The system simplifies back-end integrations because you can use `list_m2m_tokens` to retrieve necessary keys without leaving your primary workspace.

Real-World Applications

Onboarding a new enterprise client

A PM needs to onboard 'Global Corp' and give them three specific users, each with an Admin role. They prompt their agent: 'Create tenant Global Corp and provision John Doe (Admin) and Jane Smith (Read-Only).' The MCP handles the `create_tenant`, followed by two calls to `create_user` and appropriate role assignment.

Auditing an account leak

A Support Engineer receives a report of suspicious access. They ask their agent to 'Find all users in the tenant with ID tnt_123.' The MCP runs `get_tenant_details` and then uses `list_users` filtered by that specific tenant, identifying exactly who needs immediate deletion via `delete_user`.

Decommissioning a customer account

A DevOps Admin confirms a client has left. They prompt: 'Delete the tenant named OldCo.' The MCP runs `list_tenants` to find the ID, and then executes `delete_tenant`, ensuring all associated user data is cleanly removed.

Checking role assignment compliance

A Founder needs a list of every possible access level. They prompt: 'List all roles and permissions.' The MCP returns both the full set of system roles via `list_system_roles` and the granular rights available using `list_permissions`, confirming compliance.

Patterns to Avoid

Treating it like a simple database query

✗ AVOID

A user tries to use only `get_user_details('email')` and gets frustrated when the system asks for a tenant ID, thinking they need more context.

✓ INSTEAD

Remember that identity is always multi-tenant. Always start by listing all tenants using `list_tenants`, or provide the target tenant ID first to give your agent enough context.

Manually managing credentials

✗ AVOID

Copying and pasting API keys into a script every time you run an audit, which is slow and insecure.

✓ INSTEAD

The MCP handles the connection securely using client IDs. You can confirm setup status anytime with `check_environment_status`.

Trying to update roles without listing them

✗ AVOID

A user tries to assign a role they assume exists, but the agent fails because the name is incorrect.

✓ INSTEAD

Always run `list_system_roles` first. This gives you the definitive list of names and types for accurate assignment.

The Right Fit

Use this MCP if your primary headache involves managing user accounts, roles, or customer tenants across a multi-client SaaS architecture. If you need to perform lifecycle actions—like creating, deleting, or auditing access—this is the tool. Don't use it if you just need general knowledge about authentication methods; for that, you might look at tools focused on general API documentation or logging services. Use this when your goal is *actionable identity management*. If your only task is reading basic metrics (like revenue reports), this MCP isn't the right fit; you need a dedicated financial reporting tool instead.

Managing access across client accounts feels like juggling dozens of separate dashboards.

Right now, if you need to check what permissions exist for one user in one specific tenant, you have to jump through hoops. You log into the dashboard, find the correct customer account (the 'tenant'), navigate deep into settings, search for the user by email, and then click three different times just to see their role—all while worrying about which API key worked last.

With this MCP, that process vanishes. Your agent handles the complexity. You tell it: 'What is Jane Doe's access level in Acme Corp.' And you get a direct answer without opening any web UI. It turns hours of dashboard clicking into one simple command.

Frontegg MCP provides complete control over your user lifecycle.

You eliminate manual steps like checking if a tenant exists before trying to add users, or needing separate workflows for listing roles and then assigning them. The agent coordinates the entire operation: it first runs `list_tenants` to confirm the environment is ready, then uses `create_user`, and finally assigns the correct role.

The result isn't just convenience; it’s guaranteed accuracy. You get a single source of truth for every identity action you take.

Frontegg: 12 Identity Management Tools

These twelve tools allow your agent to perform every critical identity operation—from listing all users globally to deleting entire customer tenants.

#	TOOL	DESCRIPTION
01	`create_tenant`	Creates a brand new customer account (tenant) within the system.
02	`create_user`	Adds and provisions an individual user profile to the platform.
03	`delete_tenant`	Completely removes a customer account (tenant) from the system.
04	`delete_user`	Removes an existing user profile entirely.
05	`check_environment_status`	Verifies that the API connection and necessary credentials are working correctly.
06	`get_tenant_details`	Retrieves specific metadata about an existing customer account (tenant).
07	`get_user_details`	Fetches comprehensive profile information for a specific user.
08	`list_m2m_tokens`	Lists all available Machine-to-Machine tokens associated with the system.
09	`list_permissions`	Outputs a list of specific, granular permissions that can be assigned or audited.
10	`list_system_roles`	Displays all available system roles (like Admin or Read-Only) that users can be given.
11	`list_tenants`	Retrieves a complete list of every customer account currently active on the platform.
12	`list_users`	Provides a global listing of all user profiles across all tenants.

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 tenants in our Frontegg environment.

Retrieving tenants... I found 10 active customer tenants, including 'Acme Corp' (ID: tnt_123) and 'Globex Inc' (ID: tnt_456). Would you like to see the users for Acme Corp?

U Find the user details for 'jane@example.com'.

Searching users... I found Jane Doe (jane@example.com). She is assigned to tenant 'Acme Corp' and has the 'Admin' role. Her account was created on Oct 1st.

U Create a new tenant named 'Stark Industries'.

Tenant created! I have successfully provisioned the new tenant 'Stark Industries' in your Frontegg environment. The new Tenant ID is tnt_789012.

Frequently Asked Questions

01 **How does Frontegg MCP handle multi-tenancy?**

It treats each customer account as a separate, managed silo (a tenant). You can use `list_tenants` to see all accounts and then target specific tenants when running user or role commands.

02 **Can I list every single user profile using the Frontegg MCP?**

Yes. The `list_users` tool provides a global listing of users, but you can also use `get_user_details` to fetch specific profiles for auditing.

03 **What if I need to delete a user or tenant?**

The MCP supports both cleanup actions. Use `delete_user` to remove an individual profile, or `delete_tenant` to wipe out all data associated with a whole customer account.

04 **Does Frontegg MCP help me audit security roles?**

Absolutely. You can run `list_system_roles` and `list_permissions` together to get the full catalog of what is assignable across your entire platform.

05 **Do I need specific keys to use Frontegg MCP?**

Yes, you must provide a valid Client ID and API Key. You can verify that everything is connected using the `check_environment_status` tool.

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 `"frontegg": { "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

Frontegg 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 Frontegg. 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	Frontegg MCP
Server ID	`019d75a1-73f7-7022-b3fb-312b4486aa7d`
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/frontegg.