

MCP SERVER

NO CODE

CLOUD HOSTED

# Chargebee MCP

Automate your entire billing workflow via natural language.

Chargebee MCP manages complex billing cycles, letting you automate subscription changes and track revenue data directly within your AI workflow. Connect to this service to list active subscriptions, query customer profiles, or cancel accounts without leaving your agent environment. It gives operations teams full control over the entire subscriber lifecycle—from onboarding a new user to reconciling overdue invoices.

**A+** Quality Score 100/100

recurring-billing

revenue-operations

subscription-management

churn-reduction

invoicing

saas-metrics



# 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.

### 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.

### 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.

### 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**

Encryption at rest

**Ed25519**

PKI vault signatures

**24h TTL**

Ephemeral session keys

**V8 Isolate**

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.

**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.

**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.

**06 — Honeypot Trap System**

Phantom credentials are injected into isolated environments. If a honeypot 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.

**FinOps**

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

**Firewall & DLP**

PII redaction activity, sensitive data protection counters, and security event timeline.

**Agent Activity**

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

**Tool Health**

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

**Incident Log**

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

Get started at [cloud.vinkius.com](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# Chargebee MCP

10 tools available

Cloud-hosted on Vinkius

This MCP connects your AI client directly to your billing engine, giving you natural language access to every aspect of your recurring revenue model. Instead of logging into a dashboard and clicking through menus, you simply ask your agent what needs to happen.

Need to onboard a new user? Your agent can execute the necessary steps: first creating the customer profile, then setting up the initial subscription based on available plans. Need to adjust an account mid-cycle? You can inspect existing subscriptions or even cancel them entirely.

This level of control means your billing data—customer details, payment statuses, and invoice history—is immediately actionable within a single workflow. When you connect this MCP via Vinkius, you bring the full power of Chargebee directly into your chat interface, letting you keep revenue operations inside your primary AI environment.

---

## Core Capabilities

### 01 — Manage Subscription Status

List all subscriptions—active, trialing, or cancelled—and retrieve specific details about them.

### 03 — Reconcile Invoices and Payments

Query the system to find invoices that are paid, pending, or overdue, helping you track immediate cash flow.

### 05 — Update Product Offerings

List available product plans and addons to understand what you can sell or bundle with a client account.

### 02 — Track Customer Profiles

Fetch complete records for any customer, including contact information and billing payment sources.

### 04 — Onboard New Users Programmatically

Create new customer accounts and immediately establish their initial subscription plan.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/chargebee-alternative](https://vinkius.com/mcp/chargebee-alternative) — connect your AI agent in three steps.

- 01 Subscribe to this MCP and provide your specific Chargebee Site name and API Key.
- 02 Your AI agent uses these credentials to authenticate against the billing system's tools.
- 03 You instruct the agent using natural language, which translates into a structured API call (e.g., 'Find all invoices over 60 days old').

The bottom line is that your AI client handles the connection details and the necessary data calls, presenting you with clean answers without requiring manual authentication or dashboard navigation.

---

## Built For

Billing Managers, Revenue Operations Specialists, and SaaS Product Owners use this. They are stuck in repetitive cycles of checking dashboards for churn risks, reconciling overdue invoices across multiple tabs, or manually updating customer profiles because a sales deal changed scope.

### Revenue Operations Specialist

Manages the full lifecycle of subscriptions. They use this to list active subscriptions and execute complex workflows like canceling an account or creating a new one based on support ticket details.

### Billing Manager

Runs end-of-month reports on revenue health. They query the system to list invoices, filter by status (like overdue), and get detailed customer information to reconcile payment discrepancies quickly.

### Product Owner

Needs to understand your pricing structure for feature planning. They use this to retrieve all available product plans or add-ons to model new subscription tiers before a launch.

## What Changes When You Connect

- 
- 01 Stop bouncing between tabs. Need to cancel an account or update a payment source? Your agent can execute the `cancel_subscription` tool and confirm the action, all in one chat window.

---

  - 02 Slash onboarding time by automating setup. Instead of manually creating profiles, your agent can run `create_customer` followed immediately by `create_subscription`, turning conversation into operational reality.

---

  - 03 Get a clear picture of cash flow instantly. You don't have to filter massive spreadsheets; you ask the agent to `list_invoices` and get immediate counts of overdue or pending payments.

---

  - 04 Better customer data means less churn risk. Use `get_customer` to pull up payment history, contact details, and current subscription status for any client ID on demand.

---

  - 05 Pricing clarity is now instant. When a salesperson asks what tiers are available, your agent can run `list_plans` or check out the full range of product addons using `list_addons`.
- 

---

## Real-World Applications

### Handling an Immediate Churn Risk

A customer support rep needs to address a user threatening to leave. The agent checks the account details by running `get_customer`, sees their subscription is about to expire, and then uses `list_plans` to suggest upgrading them before they cancel. This keeps the revenue flow positive.

### Reconciling Q3 Payments

The finance team needs a list of all invoices that are past due for specific regional clients. The agent runs `list_invoices`, filtering by 'overdue' status, and provides the exact total amount owed, avoiding manual spreadsheet reconciliation.

### Onboarding Enterprise Clients

A sales rep closes a massive deal requiring three different service tiers. The agent first runs ``create_customer`` for the new entity, then uses ``list_plans`` to select the correct pricing, and finally executes ``create_subscription`` with all the necessary details.

### Investigating Account Errors

A user reports an incorrect charge. The agent retrieves both the subscription data using ``get_subscription`` and checks the payment history via ``list_invoices``. By comparing these two records, they pinpoint exactly when the billing error occurred.

---

## Patterns to Avoid

---

### Treating it like a spreadsheet lookup

#### X AVOID

Just dumping 10 queries to find if an invoice is overdue and what plans exist. This requires dozens of manual checks.

#### ✓ INSTEAD

Ask your agent directly: 'What are the three most urgent billing issues we have right now?' The agent intelligently runs ``list_invoices`` for overdue items, and cross-references it with current subscriptions using ``list_subscriptions``.

### Manually updating multiple systems

#### X AVOID

A client changes their credit card details. You have to log into the billing portal, update the payment method, then maybe send a follow-up email.

#### ✓ INSTEAD

Your agent can handle this workflow: first verifying your contact info with ``get_customer``, and then updating the necessary financial records through the platform's integrated tools.

### Forgetting to check product limits

#### X AVOID

Thinking you can offer a premium tier without checking if it actually exists or how it prices. You might promise something unbillable.

#### ✓ INSTEAD

Always run ``list_plans`` first. This confirms the exact pricing and features available before promising anything, ensuring your proposal is financially sound.

---

## The Right Fit

Use this MCP if your core pain point involves complex revenue operations: managing subscription lifecycles, reconciling payment discrepancies, or automating customer onboarding. If you need to perform actions like canceling a service, setting up a recurring charge, or checking the status of an overdue invoice, this is what

you need.

Don't use this if your only goal is simple data aggregation that doesn't require writing back to the system. For example, if you just want to read static marketing copy about plans, searching a public website is enough. If you simply need to query basic contact details without billing context, other specialized directory tools might be sufficient, but for anything involving money or service status, this MCP is mandatory.

---

## The administrative nightmare of monthly billing reconciliation

Every month involves the same tedious sequence: opening the dashboard to check paid vs. pending invoices, switching tabs to cross-reference customer IDs with their current subscription status, and then copying that data into a spreadsheet just to spot discrepancies. It's slow, error-prone clicking through five different views.

With this MCP, you describe the problem once in natural language. Your agent handles all those clicks and lookups internally—running tools like `list_invoices` or checking specific subscription details using `get_subscription`. You get a single, synthesized answer that tells you exactly what's wrong with your revenue, instantly.

---

## Get full control over the entire subscriber lifecycle

Previously, if a client needed to downgrade or cancel service, it was a multi-step process requiring manual intervention: find their account ID, locate the correct billing panel, and execute the cancellation. This created friction and delayed revenue action.

Now, you just tell your agent to 'cancel the subscription for this user.' The MCP handles the entire sequence—it verifies the customer via `get_customer`, and then runs `cancel_subscription` directly. It's that simple.

---

# Chargebee MCP: 10 Tools for Revenue Operations

These tools give you direct, conversational access to every core function of your billing system—from listing invoices to managing entire subscriber lifecycles.

#	TOOL	DESCRIPTION
01	<code>cancel_subscription</code>	Cancels an existing customer's subscription account.
02	<code>create_customer</code>	Creates a brand new profile for a paying customer.
03	<code>create_subscription</code>	Sets up a paid, recurring subscription for an existing or new customer.
04	<code>get_customer</code>	Retrieves all specific details associated with one customer ID.
05	<code>get_subscription</code>	Fetches the full data sheet for a single subscription record.
06	<code>list_addons</code>	Shows all additional product features or upgrades that can be sold.
07	<code>list_customers</code>	Provides an overview list of all customers in your system.
08	<code>list_invoices</code>	Lists billing invoices, allowing filtering by status or customer ID.
09	<code>list_plans</code>	Shows the full catalog of standard product pricing plans available for sale.
10	<code>list_subscriptions</code>	Lists all subscriptions, supporting filters like status or date range.

---

## See It in Action

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

### **U** List all active subscriptions in Chargebee.



I've fetched the subscriptions. You have 12 active subscriptions, including 'sub\_123' (Premium Plan) and 'sub\_456' (Standard Plan).

### **U** Show me details for customer 'cust\_789'.



Customer 'cust\_789' is John Doe (john@example.com), based in New York. They have an active payment source on file.

### **U** Find all overdue invoices.



I've analyzed your invoices. There are currently 3 overdue invoices totaling \$450. Would you like the list of customer IDs for these invoices?

---

## Frequently Asked Questions

### **01** How do I find out what plans are available using Chargebee MCP?

Use the `list\_plans` tool to view all standard product pricing tiers. This gives you a comprehensive catalog of services and their base costs.

### **02** Can I use Chargebee MCP to create an account from scratch?

Yes, you can execute this in two steps: first, run `create\_customer` to generate the profile. Then, immediately follow up with `create\_subscription` to activate their paid service.

**03 What if I need to check all overdue invoices using Chargebee MCP?**

You simply ask your agent to list invoices and specify 'overdue' in the request. The system runs `list\_invoices` and provides a filtered, actionable summary of what needs immediate attention.

---

**04 Is Chargebee MCP only for existing subscribers?**

No, it supports both new and existing accounts. You can create entirely new customer profiles using `create\_customer` before setting up their first paid subscription.

---

**05 Does Chargebee MCP help with product add-ons?**

Yes, the `list\_addons` tool shows you all available extra features or upgrades. You can use this information to upsell clients or structure new bundles for them.

---







---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"chargebee-alternative": {   "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	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

# Chargebee 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](https://cloud.vinkius.com) →

[vinkius.com](https://vinkius.com) · [support@vinkius.com](mailto: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 Chargebee. 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	Chargebee MCP
Server ID	019dd0cb-7c0f-72c5-ae57-003baff94927
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### 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/chargebee-alternative](https://vinkius.com/mcp/chargebee-alternative).