

MCP SERVER

NO CODE

CLOUD HOSTED

Recharge MCP

Automate every billing and recurring subscription task in conversation.

Recharge MCP handles everything related to recurring billing and subscriptions. Use this connector to manage customer profiles, track order history, analyze revenue charges, and modify subscription status entirely through conversation. It's the fastest way to get deep operational data from your e-commerce billing system without touching a dashboard or running complex scripts.

A+ Quality Score 100/100

subscriptions

recurring-billing

shopify-subscriptions

ecommerce-automation

recharge



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.

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 — connect your AI agent in under 60 seconds.

Recharge MCP

11 tools available

Cloud-hosted on Vinkius

Managing recurring revenue is messy. You shouldn't have to jump between the Recharge dashboard, write API calls, and cross-reference billing records just to know if a customer's payment failed or what their current plan details are. This MCP lets your AI client handle all that complexity using natural language. Need to see every active subscription? Just ask. Want to find out why revenue dropped last month by analyzing charges? Ask the agent to pull that data, and it'll surface the financial insights for you. The goal is simple: let your agent perform complex billing operations—from listing customer addresses to updating a plan quantity—all in one chat window. By connecting this MCP through Vinkius, you get immediate, conversational access to all 11 core tools needed to keep subscription commerce running smoothly.

Core Capabilities

01 — Analyze and list active subscriptions

You can pull a full list of existing plans or retrieve the detailed metadata for any single subscription.

03 — Track payments and order history

Get details on specific orders or pull comprehensive lists of processed billing charges to understand revenue flow.

05 — Handle customer lookups and list products

Search your database to pull specific customer records or see all available subscription product offerings.

02 — Manage customer data and addresses

Query subscriber profiles, get specific billing information, and list all associated shipping addresses.

04 — Update subscription status and quantity

Programmatically change the number of items in a plan or modify an existing subscription's parameters without manual dashboard clicks.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to the Recharge MCP on Vinkius.
- 02 Input your API Access Token, which you get from your store's settings.
- 03 Start chatting with your AI client and ask for billing or customer data using plain language.

The bottom line is: once connected, your agent handles all the complex back-end calls so you don't have to think about them.

Built For

This MCP is for e-commerce operations teams and billing analysts who spend too much time jumping between dashboards and running repetitive API checks. If your job involves ensuring customers are billed correctly or diagnosing a recurring revenue issue, this connector saves hours of manual effort.

E-commerce Operations Manager

You use the agent to check all customer records and verify subscription statuses when onboarding new merchants.

Billing Analyst

You run reports on processed charges or list billing charges to reconcile revenue differences against expected payments.

Product Manager (Subscription Focus)

You use the tools to list all available subscription products and check if new product metadata is correct before launch.

What Changes When You Connect

- 01 Stop cross-referencing tabs. Instead of manually checking customer records, you can ask the agent to use `get_customer` and instantly pull profiles and mailing addresses into your chat.

-
- 02** Never miss a payment detail again. You can list all billing charges using `list_billing_charges`, giving you immediate visibility into historical revenue streams without running reports.
-
- 03** Manage plans on the fly. If a customer needs more items, use `update_subscription` to modify their plan quantity or change the next charge date—no dashboard clicks needed.
-
- 04** Keep track of everything from sign-up to delivery. You can list all subscription orders (`list_subscription_orders`) and get specific order details using `get_order`, keeping your fulfillment pipeline tight.
-
- 05** Quickly identify service gaps by running a full review with `list_subscriptions` or seeing what products are available via `list_products`.
-

Real-World Applications

Diagnosing revenue drops for finance

The billing analyst notices an unexpected dip in projected revenue. They ask the agent to list all billing charges and get a full overview of subscriptions using `list_subscriptions` to pinpoint exactly which product line is underperforming.

Processing bulk account changes

The product team launches a new tier that requires changing 50 existing plans. They ask the agent to list all subscriptions, filter for the target group, and then loop through calling `update_subscription` for each one.

Updating a customer's shipping details

The operations manager needs to send an address update for a high-value client. Instead of finding the customer in multiple tabs, they ask the agent to use `get_customer` and then call `list_shipping_addresses` to get the right record.

Handling payment failures

A customer reports a failed charge. The agent pulls the specific order details using `get_order`, checks the billing charges via `list_billing_charges`, and determines if an update is needed, guiding the user to fix it.

Patterns to Avoid

Using general database queries

X AVOID

Trying to write a complex SQL-like query in your agent to 'find all active plans for customer X that charged over \$10 last month.'

✓ INSTEAD

Don't write the query. Just ask: 'Show me the charges and subscriptions for this customer.' The agent handles combining the necessary calls using ``get_customer`` and ``list_billing_charges``.

Copy-pasting from the dashboard

X AVOID

Manually copying a list of 100 product IDs, then pasting them into a script to check status.

✓ INSTEAD

Instead, ask the agent: 'Give me an inventory snapshot.' The tool calls ``list_products`` and gives you structured data directly in your chat.

Calling updates without confirmation

X AVOID

Asking the agent to run ``update_subscription`` immediately after finding a customer, risking changing the wrong thing.

✓ INSTEAD

Always confirm first. Ask: 'First, use ``get_subscription_details`` for Plan ID 123. Once I confirm this is correct, then update it.' This prevents accidental changes.

The Right Fit

Use this MCP if your workflow revolves around understanding the financial state of recurring revenue and managing subscriber accounts. You need to read charges (`list_billing_charges`), check customer profiles (`get_customer`), or actively change a plan's status (`update_subscription`). Don't use it if you primarily just need basic inventory counts (a general catalog tool is better) or if your core task involves non-financial data, like tracking physical shipping logistics outside of the billing system. If all your questions relate to 'who owes money,' 'what plan they are on,' or 'was that charge processed?' then this MCP is exactly what you need.

The Billing Dashboard Maze

Today, checking a customer's status requires clicking through five different tabs: the main subscription view, their address book, the order history page, the billing charges log, and finally, digging into the product catalog to see what they actually bought. You then have to copy all that data—the plan name, the charge amount, the last updated date—and paste it into a separate sheet just for analysis.

With this MCP, you tell your agent exactly what you need. You don't navigate; you ask. The agent uses its tools behind the scenes to pull together the customer profile, their current plan status, and all associated billing charges into one simple conversational response. It brings clarity instantly.

Recharge MCP: Billing Data at Your Fingertips

You no longer need to run separate API calls for every data point. If you want the current subscription details, you call `get_subscription_details`. If you're troubleshooting an old payment issue, you just ask for a list of billing charges using `list_billing_charges`.

Everything is connected conversationally. The result isn't a JSON payload; it's a direct answer that tells you exactly what to do next.

Recharge MCP: 11 Tools for Billing & Orders

These tools let you interact with every facet of your subscription commerce system—from listing customers to updating complex billing parameters.

#	TOOL	DESCRIPTION
01	<code>cancel_subscription</code>	Immediately cancels an existing customer's subscription plan.
02	<code>get_customer</code>	Retrieves full profile details for a specific subscriber ID.
03	<code>get_order</code>	Pulls the complete details for a single, processed order.
04	<code>list_products</code>	Gets an inventory list of all subscription products available in your store.
05	<code>get_subscription_details</code>	Retrieves comprehensive status and metadata for one specific active subscription.
06	<code>list_shipping_addresses</code>	Lists all shipping addresses tied to a customer profile.
07	<code>list_billing_charges</code>	Gathers historical data on all processed financial charges.
08	<code>list_customers</code>	Provides a list of every subscriber currently in the Recharge system.
09	<code>list_subscription_orders</code>	Retrieves a history of all processed subscription orders.
10	<code>list_subscriptions</code>	Gives you an overview list of every current and canceled subscription plan.
11	<code>update_subscription</code>	Changes the parameters, quantity, or scheduled charge date for a live subscription.

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 Recharge.



I've retrieved your subscriptions. You have 50 active subscriptions, including 'Premium Coffee Plan' and 'Monthly Supplement Pack'.

U Show me all active subscriptions that are due for renewal in the next 7 days.



142 subscriptions due for renewal within 7 days. Tomorrow: 23 renewals (\$4,567 total). Wednesday: 31 renewals (\$6,234). Thursday-Sunday: 88 renewals (\$18,900). Top products: "Monthly Coffee Box" (45 renewals), "Protein Pack" (34), "Wellness Bundle" (28). 8 subscriptions have failed payment methods that need updating. 3 customers requested skipping their next delivery. Total projected renewal revenue: \$29,701.

U Show me the churn analysis for the last 30 days with cancellation reasons.



89 cancellations in the last 30 days (4.2% churn rate, down from 5.1% last month). Reasons: "Too expensive" (34%), "No longer needed" (22%), "Switching to competitor" (18%), "Product quality" (12%), "Delivery issues" (8%), "Other" (6%). Average subscription length before cancellation: 4.7 months. Revenue lost: \$12,340. 23 cancellations were win-back eligible. 8 customers paused instead of cancelling after retention offers.

Frequently Asked Questions

01 How can I check if a customer's payment failed using Recharge MCP?

You can use ``get_order`` and ``list_billing_charges``. The agent pulls the most recent order details and cross-references them with your billing charges to spot discrepancies or failures.

02 What is the best way to see all products available for subscription?

Use the `list_products` tool. It gives you a complete inventory snapshot of every product that can be sold as a recurring subscription service.

03 Can I change a customer's plan without using the website dashboard?

Yes, use the `update_subscription` tool. You simply tell your agent which subscription you want to modify and what changes to make—like increasing quantity or changing the date.

04 What if I need to find out all my customers' names?

The `list_customers` tool provides a comprehensive list of every subscriber currently registered with Recharge, giving you an immediate roster.

05 Does the MCP help me track where customers are located?

Use `list_shipping_addresses`. This tool retrieves all saved shipping addresses linked to any specific customer profile in your system.

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 `"recharge": { "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

Recharge 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 · 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 Recharge. 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	Recharge MCP
Server ID	019dd148-e16e-7123-9a8d-74bb122ef7c0
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/recharge.