MCP SERVER | NO CODE | CLOUD HOSTED

Checkout.com MCP for AI Agents

Manage global payment transactions and refunds from any chat interface

Checkout.com MCP lets your AI client manage global payments conversations. You can track transactions, capture authorizations, and process refunds across 150+ currencies without leaving your chat window.

F Quality Score 3.6/100

global-payments | transaction-processing | refunds | payment-gateway | currency-management

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

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.

Checkout.com MCP

8 tools available
Cloud-hosted on Vinkius

Managing global e-commerce payments used to mean logging into a dashboard, navigating between tabs, and manually cross-referencing transaction IDs. Now, you talk to your AI client and get the answers instantly.

This MCP lets you take full control of every aspect of your payment lifecycle in natural conversation. Need to know what happened to an authorization? Ask. Want to process a refund for a specific order ID? Tell it to do it. You can list recent payments, check detailed history via `list_checkout_payments`, or even verify webhook delivery using `list_checkout_webhooks`.

It handles the messy details too; you don't have to worry about minor unit math because we manage precise financial amounts in minor units for high-accuracy transactions. This kind of deep control over global payments is usually reserved for complex internal tooling, but connecting through the Vinkius catalog makes it accessible right from your workspace. You can even perform mutable operations like capturing or voiding funds directly, giving you complete oversight and auditability.

Core Capabilities

01 — List all recent payments

Retrieves a list of transactions that have recently been processed through Checkout.com.

02 — Get detailed payment history

Fetches comprehensive information for a single, specific transaction ID.

03 — Process refunds or void payments

Issues immediate commands to refund captured funds or cancel pending authorizations.

04 — List payment actions history

Shows the full timeline of events—like authorization, capture, and refund—for any given payment ID.

05 — Retrieve core account data

Pulls foundational details about your Checkout.com account and user settings.

06 — Monitor webhooks

Reviews which system hooks are configured to ensure automated integrations are running correctly.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Checkout.com Secret Key and API Prefix.

02 Connect the MCP to any compatible AI client (like Cursor or Claude).

03 Ask your agent a question, like 'What happened to payment X?' and it executes the necessary actions using the toolset.

The bottom line is: you never have to leave your chat interface to manage payments; your AI agent handles all the API calls for you.

Built For

This MCP is built for anyone who spends time reviewing payment statuses, auditing transactions, or needing quick financial data. If you're a developer verifying webhooks or an e-commerce manager processing refunds at 10 PM, this tool saves clicks.

E-commerce Operations Manager

Needs to audit payment statuses and process retroactive refunds for orders without having to log into the main financial dashboard.

Finance Analyst

Requires quick, conversational access to global transaction volumes and full lifecycle histories across multiple currencies for reconciliation.

Backend Developer

Needs to verify webhook delivery or pull core account info during API integration testing without running manual scripts.

What Changes When You Connect

01 Audit transaction history instantly: Check the full lifecycle of any payment using `list_payment_actions` without digging through multiple screens.

02 Control payments conversationally: You can capture, void, or refund funds simply by asking your agent to perform the action.

03 Maintain high accuracy with minor units: The MCP handles precise financial amounts in minor units, ensuring zero calculation errors for complex accounting.

04 Verify integrations quickly: Use `list_checkout_webhooks` to confirm that automated data feeds are running flawlessly during development or maintenance.

05 Get account health checks: Quickly retrieve core user and account information using `get_checkout_account_info`, keeping your operational context fresh.

06 Stay in one place: Process payments, view details (`get_payment_details`), and list transactions all within a single chat window.

Real-World Applications

Investigating a chargeback status

A customer service agent asks their AI client, 'What was the full history for order ABC?' The agent uses `list_payment_actions` and immediately provides the detailed timeline showing authorization, capture, and any subsequent adjustments.

Processing a bulk refund request

The finance team needs to issue refunds for several cancelled orders. Instead of logging into the portal multiple times, they instruct their agent to `refund_checkout_payment` using the list of payment IDs provided.

Checking API connectivity post-update

A developer needs to confirm that a new webhook listener is active. They ask the agent to `list_checkout_webhooks`, instantly validating the configuration against Checkout.com's live settings.

Confirming payment readiness

An e-commerce manager needs to ensure an authorization can be finalized before shipping goods. They use the MCP to `get_payment_details` and confirm the status, then instruct the agent to `capture_checkout_payment`.

Patterns to Avoid

Treating payments like simple database records

✗ AVOID

Thinking you just need a list of payment IDs. You run 'list recent payments' but don't know if they are authorized, captured, or voided.

✓ INSTEAD

Always use `get_payment_details` and then follow up with `list_payment_actions`. This gives the full story: what happened, when it was done, and why. It prevents assumptions about status.

Manually recalculating funds

✗ AVOID

Trying to figure out the refund amount by cross-referencing multiple reports across different systems.

✓ INSTEAD

The MCP handles this. Use `get_payment_details` first, then use `refund_checkout_payment`. The tool manages the minor units and currency logic so you don't have to.

Ignoring webhook health

✗ AVOID

Assuming that because a payment succeeded in your system, Checkout.com was notified. This leads to silent data gaps.

✓ INSTEAD

Before launch or after updates, always run `list_checkout_webhooks` to verify the setup. If it looks wrong, you know where to focus your debugging efforts.

The Right Fit

Use this MCP if your core job revolves around financial reconciliation, payment status monitoring, and managing transaction lifecycles in e-commerce. You need conversational access to perform actions like refunding funds or capturing payments on Checkout.com.

Don't use it if you only need simple read access (like viewing general account info) but don't require the ability to change state (e.g., you just want a report). For purely passive reporting, an API wrapper might suffice. But if your workflow requires taking action—voiding money or confirming fund availability—this MCP is what you need. If your domain is outside of global payment processing (like HR records or inventory counts), this won't help.

Checkout.com MCP for AI Agents: Solving Payment Reconciliation Pain

Right now, reviewing a single customer transaction means jumping between the merchant portal, your internal accounting software, and potentially a webhook log just to piece together what happened. You pull up payment IDs, check statuses, and copy data—it's slow, error-prone clicking.

With this MCP, you ask your agent about any transaction. It automatically compiles all necessary steps: it pulls the initial details, checks the action history (`list_payment_actions`), and tells you exactly where the money is in the lifecycle. You get a complete audit trail instantly.

Checkout.com MCP for AI Agents: Managing Global Currency Flows

Handling refunds or captures across 150+ currencies requires meticulous attention to minor units and exchange rates, making manual

This MCP manages those complex financial details for you. You simply request an action—like refunding funds—and the agent executes it with

intervention a massive headache. One slip up, and the accounting gets messy.

high accuracy, letting you focus on the customer, not the math.

Checkout.com: 8 Tools for Payment Transaction Operations

Use these tools to perform every necessary action on a transaction, from listing recent payments to capturing authorized funds and managing webhooks.

#	TOOL	DESCRIPTION
01	`capture_checkout_payment`	Finalizes an authorized payment, converting it into captured funds.
02	`get_checkout_account_info`	Retrieves fundamental details about your Checkout.com account and user settings.
03	`list_payment_actions`	Generates a full chronological list of every event that occurred with a specific payment ID.
04	`get_payment_details`	Gathers all available data points for one particular transaction record.
05	`list_checkout_payments`	Pulls a list of the most recent payments processed through your account.
06	`list_checkout_webhooks`	Shows all webhooks currently set up, verifying system connectivity and data flow.
07	`refund_checkout_payment`	Initiates the process of returning money to a payment that was previously captured.
08	`void_checkout_payment`	Cancels an authorization before it has been captured or processed.

See It in Action

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

U What were all the payments processed last week that are currently only 'Authorized'?

Payment Summary: Authorized Funds (Last 7 Days)

ORDER ID	AMOUNT	STATUS	DATE
#90210	$54.99	Authorized	Yesterday
#33876	€12.50	Authorized	3 Days Ago

Action required: These payments need to be captured before fulfillment.

U I need to issue a refund for payment ID 'pay_99a'. It was for $75 last month.

Refund Request Status

✅ Refund of **$75.00** successfully submitted for `pay_99a`.

The funds are being processed and the new status will be reflected in the payment actions within minutes. Please check back later to confirm the final reversal details.

U Can you list my recent payments, but only show me those that failed or were voided?

Voided & Failed Payments Report

Here are the three transactions matching your criteria:

Order #XYZ: $12.00 (Status: Voided) - Action taken manually.*
Order #ABC: €5.00 (Status: Declined) - Reason: Insufficient funds.*
Order #DEF: $9.99 (Status: Failed) - Attempt reason: Invalid card number.*

Frequently Asked Questions

01 **How do I check the full history of a payment using Checkout.com MCP for AI Agents?**

You simply ask your agent what happened to a specific transaction ID. It retrieves all associated lifecycle actions, showing the exact dates and times when the payment was authorized, captured, or refunded.

02 **Can I use Checkout.com MCP for AI Agents to process refunds?**

Yes. You can instruct your agent to refund a previously captured payment. It handles the required financial logic and submits the request directly through Checkout.com's secure channels.

03 **Is this better than checking the Checkout.com dashboard manually?**

It's faster and more conversational. Instead of clicking through multiple reports, you ask your agent for exactly what you need—a summary, a list, or an action—and get it immediately.

04 **Does Checkout.com MCP for AI Agents handle different currencies?**

Yes. It is built to manage global payments across over 150 currencies, ensuring the math and financial units are precise no matter where the transaction originated.

05 **If I need to void a payment, do I have to use the website?**

No. You can ask your AI agent to perform the void action directly through the MCP. It verifies the status first and then executes the cancellation command for you.

06 **What if my webhook setup is broken? Can Checkout.com MCP help?**

You can use the MCP to list and review your configured webhooks. This lets you quickly verify connectivity details, helping a developer pinpoint exactly where an integration failed.

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 `"checkoutcom": { "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

Checkout.com 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 Checkout.com. 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	Checkout.com MCP
Server ID	`019d756e-81b2-7026-a29d-c1468c0140bd`
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/checkoutcom.