

MCP SERVER

NO CODE

CLOUD HOSTED

# Instamojo MCP

Track payments, manage refunds, and automate payouts.

Instamojo MCP connects your AI agent directly to India's payment infrastructure. This tool lets you automate financial monitoring, process transactions, and track funds across major e-commerce operations in the country. Your agent can fetch specific payments, list all recent payouts, manage refund details, or check active disputes—all without manual logins or API calls. It's designed for anyone managing high-volume commerce that needs reliable access to payment data.

**A+** Quality Score 98.33/100

payment-gateway

refund-management

payouts

dispute-resolution

transaction-tracking

indian-fintech



# 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.

# Instamojo MCP

10 tools available

Cloud-hosted on Vinkius

Instamojo lets your AI client handle the full lifecycle of e-commerce finance in India. Instead of manually logging into a dashboard, you tell your agent what financial information you need, and it pulls it directly from the Instamojo API. You can automatically check incoming payments to see if they cleared, track exactly where payouts are heading, or pull up records for refunds that went through.

This is ideal for automating daily financial checks and building robust back-office processes. It moves beyond simple data retrieval; you can use your agent to monitor disputes in real time and manage payment requests across different systems. Because Vinkius hosts this MCP, you connect once from any compatible client—like Cursor or Claude—and get access to all these crucial tools. You'll spend less time tracking down transaction IDs and more time making business decisions.

---

## Core Capabilities

### 01 — Retrieve Payment Details

Fetch the specific information for any single payment record.

### 02 — Manage Transaction Requests

Get details on pending or scheduled payments and payment requests.

### 03 — Track Refunds and Payouts

Check the status of processed refunds, view all payout records, and list historical payouts to bank accounts.

### 04 — Monitor Financial Health

List all payments received or track generated payment links for e-commerce campaigns.

### 05 — Resolve Disputes

Pull lists of both active and past financial disputes to help with reconciliation.

# One Click on Vinkius — From Prompt to Execution

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

- 01 Your agent determines the required action, for example, 'I need to know all payouts from last week.'
- 02 It calls the appropriate tool within this MCP, sending only the necessary identifiers (like a date range or transaction ID).
- 03 The Instamojo API executes the request and sends back clean, structured data that your agent can then read aloud or pass directly to another system.

The bottom line is you give instructions in plain English, and this MCP handles all the complex financial data fetching behind the scenes.

---

## Built For

The e-commerce operations manager who gets sick of checking ten different tabs just to reconcile a single day's sales. It's for finance analysts and business owners whose job depends on accurate, real-time payment data flow from India.

### **E-commerce Operations Manager**

Uses this MCP daily to verify that payments received match the orders placed, quickly checking if funds are available or if a dispute needs immediate attention.

### **Finance Analyst**

Relies on the tool to pull historical payout and refund data for month-end reporting, drastically reducing reconciliation time.

### **Business Owner**

Uses it to get a quick overview of financial health—seeing how many payments came in versus what was paid out or disputed—without deep technical knowledge.

## What Changes When You Connect

- 
- 01 Stop manually checking dashboards for transaction status. By using `get_payment` and `list_payments`, your agent instantly checks the current state of any payment so you know exactly when funds hit.

---

  - 02 Keep track of money that hasn't cleared yet. You can use `list_payment_requests` to see pending payments and `get_payment_request` to check on specific requests without logging in.

---

  - 03 Simplify returns management. Instead of digging through emails, your agent uses `get_refund` or `list_refunds` to pull up the full details of any refund transaction instantly.

---

  - 04 Never worry about lost funds again. Monitoring disputes with `list_disputes` means you know immediately if there's an issue that needs human review.

---

  - 05 Build better reports for finance teams. Using `list_payouts` and `list_payments` lets your agent pull all the necessary figures to reconcile accounts automatically.
- 

---

## Real-World Applications

### Reconciling end-of-week sales figures

The operations manager needs to prove that \$50,000 was collected this week. They tell their agent: 'List all payments from the last seven days.' The agent uses `list_payments` and then checks specific details using `get_payment` for verification.

### Handling a large batch of refunds

A client needs to process 20 returns. They instruct their agent: 'List all refund IDs from the last month.' The agent uses `list_refunds`, then runs `get_refund` on each ID to confirm the status before emailing the client.

### Checking bank payout schedules

The business owner needs to know when funds will arrive. They ask their agent: 'What payouts are scheduled for this week?' The agent uses `list_payouts` and checks any pending items with `list_payment_requests`.

### Investigating a suspicious transaction

A dispute comes in over \$5,000. The finance analyst asks the agent to 'Check all disputes related to payment ID XYZ.' The agent uses `list_disputes` and then gets full context using `get_payment`.

---

## Patterns to Avoid

---

### Treating payments like simple data reads

#### X AVOID

Manually pulling a report that shows 'Total Payments' but doesn't show the status of payouts or refunds, leaving gaps in reconciliation.

#### ✓ INSTEAD

Don't just rely on totals. Use `list_payments` for raw income and pair it with `list_payouts` to see exactly what left the account. Always cross-reference using `get_payment`.

### Assuming all funds are settled

#### X AVOID

Thinking a payment is 'complete' just because the order was placed, without checking if the gateway has finalized the transfer.

#### ✓ INSTEAD

Always check the status. Use `list_payments` to get the initial record, but follow up with `get_payment` or `get_refund` for absolute confirmation of funds movement.

### Ignoring scheduled tasks

#### X AVOID

Failing to account for payments that are pending approval, assuming they have already cleared into the bank.

#### ✓ INSTEAD

Before counting revenue, check `list_payment_requests`. This ensures you're only reporting on settled funds and not just promises of money.

## The Right Fit

Use this MCP if your core business function requires real-time oversight of the Indian payment lifecycle—specifically managing incoming payments, processing refunds, or tracking payouts to banks. If you need general financial statements that don't rely on specific transaction IDs (like a global balance sheet), this might be too granular. Don't use it if your primary goal is simply sending out

emails; for messaging, look into dedicated communication tools. However, if the message *is* about payment status, refunds, or disputes, this MCP provides the necessary deep data access. It's built to answer 'Where exactly did that money go?'

---

---

## The mess of reconciling payments across different platforms is exhausting.

Today, checking your sales figures means logging into Instamojo, then opening a separate tab for refunds, and finally switching to another dashboard just to see the payout schedule. You spend minutes clicking through transaction IDs, copy-pasting amounts, and cross-referencing dates, all just to build one accurate picture of cash flow.

With this MCP, your agent handles that entire process in natural conversation. Instead of juggling dashboards, you ask for a single view —'What happened with the \$100 refund from last week?' And you get an immediate, structured answer.

---

## Instamojo MCP: Getting real-time payment and dispute visibility.

The tedious parts that disappear are the manual lookups of every single transaction ID. You don't have to pull up a list, then click into each one, and copy out data points; your agent handles all those steps for you.

Now, financial monitoring is conversational. Your AI client acts as an expert assistant that never gets tired, instantly giving you the status of payments or payouts without requiring a single manual click.

---

# Instamojo MCP: 10 Tools Available

Use these tools to retrieve specific payment details, list transactions, monitor disputes, and automate financial reporting using your AI agent.

| #  | TOOL                               | DESCRIPTION   |
|----|------------------------------------|---|
| 01 | <code>get_payment</code>           | Retrieves detailed information for a single, specific payment transaction.    |
| 02 | <code>get_payment_request</code>   | Pulls the current status and details of a pending payment request.            |
| 03 | <code>get_refund</code>            | Shows the specific data related to an already processed refund.               |
| 04 | <code>list_disputes</code>         | Provides a list of all financial disputes, whether they are active or closed. |
| 05 | <code>list_gateways</code>         | Lists every payment gateway that has been configured in your account.         |
| 06 | <code>list_links</code>            | Retrieves a list of all generated, shareable payment links you've created.    |
| 07 | <code>list_payment_requests</code> | Lists every pending or scheduled payment request that has been set up.        |
| 08 | <code>list_payments</code>         | Retrieves a comprehensive list of all payments received through the platform. |
| 09 | <code>list_payouts</code>          | Lists every payout transaction made to your linked bank account.              |
| 10 | <code>list_refunds</code>          | Provides a full list of all refunds that have been processed over time.       |

---

## See It in Action

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

**U** List all recent payments in Instamojo.



I'll fetch your recent transaction history from Instamojo.

**U** Show me the status of my recent payment requests.



I'll retrieve the list of payment requests and their current statuses.

**U** Check for any active disputes in my account.



I'll look up the current disputes in your account.

---

## Frequently Asked Questions

### 01 How do I check if a payment went through using Instamojo MCP?

You use `get_payment` to retrieve all the specific details for any transaction. This lets you confirm the amount, date, and final status right away.

### 02 Can I track refunds across multiple orders with Instamojo MCP?

Yes, you can list all historical refunds using `list_refunds`, or get details on a specific one using `get_refund`. This makes tracking returns straightforward.

---

**03 What is the best way to monitor pending payouts with Instamojo MCP?**

Use `list_payouts` for past payments and check `list_payment_requests` if you need to see what's scheduled or waiting to be sent out.

---

**04 Does Instamojo MCP help me find disputes?**

Absolutely. You use `list_disputes` to get a full overview of all active and past financial disagreements, helping you address them quickly.

---

**05 Can I list payment links created in Instamojo using this MCP?**

Yes, the `list_links` tool lets your agent retrieve every generated payment link, which is useful for marketing or auditing purposes.







---

# 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>"instamojo": { "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

# Instamojo 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 Instamojo. 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 | Instamojo MCP   |
| Server ID  | 019d75ba-0b8f-73a6-b01b-f1a1675ba9e0  |
| 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/instamojo](https://vinkius.com/mcp/instamojo).