MCP SERVER | NO CODE | CLOUD HOSTED

Mercury MCP

Instantly monitor cash flow and transactions.

Mercury MCP connects your AI agent directly to your Mercury business banking account. Check real-time cash positions across all accounts, pull filtered transaction history, and manage payment recipients without ever opening a browser dashboard. It gives your workflow immediate access to core startup financial data.

A+ Quality Score 100/100

business-banking | transaction-history | account-balances | startup-finance | treasury-management

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.

Mercury MCP

10 tools available
Cloud-hosted on Vinkius

Connect your company's finances conversationally. This MCP lets your AI agent handle tasks that used to require logging into the Mercury web portal. You can ask for an instant snapshot of all active account balances or pull transaction records, filtering them by dates or specific amounts right from your development environment. Need to make a payment? Your agent manages everything from listing existing recipients to creating new ones for ACH and wire transfers. Because this integration is hosted on the Vinkius catalog, you connect once using your preferred AI client—whether that's Cursor, Claude, or Windsurf—and gain access to banking data across multiple systems. You don't just get data; you execute operations. You can retrieve key account details like routing numbers and also pull full bank statements for auditing purposes.

Core Capabilities

01 — Check all cash positions

Instantly list current balances across every linked Mercury bank account.

02 — Analyze spending patterns

Pull and filter transaction records by date, amount, or the counterparty involved in the payment.

03 — Manage payments

List existing payees and create new recipients needed for ACH or wire transfers.

04 — Retrieve account fundamentals

Get specific bank metadata, like routing numbers and full account details, programmatically.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the Mercury integration via your Vinkius client connection.

02 Second, generate an API token from your Mercury dashboard settings. This key authorizes your agent.

03 Third, prompt your AI agent with a natural language request—like 'What was our spending over $5k last month?'—and it executes the necessary calls.

The bottom line is that you tell your agent what financial data or action you need, and it handles the secure connection to Mercury on your behalf.

Built For

Anyone who needs quick, reliable access to core company finances without switching between 10 different web tabs. This is for Founders, CFOs, or Ops Engineers who need cash flow visibility immediately.

Startup Founder

Needs an instant, conversational summary of the current runway and total cash position during a quick call with investors.

Chief Financial Officer (CFO)

Monitors cross-account cash flow and payment activity across multiple Mercury accounts from a single command line or development environment.

Operations Engineer

Automates daily balance checks and transaction reconciliation, pulling necessary data directly into an internal dashboard or script.

What Changes When You Connect

01 Instant Cash Visibility: Instead of logging into a web dashboard, you ask your agent for the 'total cash position' and get an immediate sum across all accounts using list_accounts and get_treasury_balance.

02 Deep Transaction Filtering: You don't just see a feed; you can prompt the agent to pull transactions above $5k from this month, instantly isolating key expenses via list_transactions.

03 Payment Workflow Automation: To send money, your agent handles listing payees with list_recipients and creating new ones using create_recipient—all without manual copy-pasting of routing numbers.

04 Audit Trail Access: Need proof of funds? You can request a full bank statement by calling list_statements, giving you structured data for accounting reports.

05 Contextual Data Retrieval: Beyond balances, the get_account tool lets your agent fetch crucial metadata like account and routing numbers when building scripts or integrating systems.

Real-World Applications

The Investor Update

A founder needs to know their current cash runway for an investor call. They prompt the agent, and it uses list_accounts to report the total funds instantly, avoiding a stressful switch to the web portal.

Monthly Expense Review

The CFO asks the agent to find all payments over $10,000 in October. The agent runs list_transactions and filters the results, providing an immediate summary of major expenditures for review.

Setting Up a Vendor Payment

The Ops Engineer needs to pay a new vendor. Instead of manually finding their details, they ask the agent to create_recipient, and it handles gathering and saving all necessary ACH information.

Debugging Financial Code

A developer needs to verify the account number for a payment script. They use get_account to pull the exact metadata needed, ensuring their code uses the correct financial identifiers.

Patterns to Avoid

Manual Dashboard Diving

✗ AVOID

A user has to log into Mercury, navigate to 'Transactions,' apply date filters, and then manually calculate a total sum of payments for their report.

✓ INSTEAD

You simply ask your agent to list_transactions filtered by the date range. The tool executes the query and provides the data set or summary total instantly.

Copying Account Details

✗ AVOID

A team member has to find the correct routing number for a payment, navigate deep into settings, and manually copy-paste it into a spreadsheet.

✓ INSTEAD

You let your agent use get_account to pull the necessary account metadata directly, ensuring perfect accuracy in your records.

Sending Payments Blindly

✗ AVOID

Attempting to send money without first confirming that the recipient exists and has valid payment details.

✓ INSTEAD

First, use list_recipients to check the saved payees. If the payee is new, you must run create_recipient before sending any funds.

The Right Fit

Use this MCP if your workflow requires real-time access to structured financial data—things like current balances (list_accounts), transaction history (list_transactions), or payment

metadata. It is essential for building automated compliance checks, cash flow models, or internal auditing tools that need source truth from the bank.

Don't use this if your core task is general document storage, project management, or writing marketing copy. If you only need to retrieve a generic text document (like an invoice PDF), there are better file-storage connectors. This MCP is purely for operational banking data access and executing financial commands.

The Financial Dashboard Click-Through

Today, getting a full picture of your company's cash requires logging into the Mercury dashboard, clicking between 'Balances,' then 'Transactions,' and finally opening separate pages to check account numbers. You spend five minutes just navigating tabs and applying filters before you even see the data.

With this MCP, you tell your agent what financial view you need. It handles the multi-step process behind the scenes, pulling specific records—like running list_transactions or fetching get_treasury_balance—and handing you a clean, summarized answer in seconds.

Getting Payment Data with Mercury MCP

Manual payment setup involves multiple steps: finding the vendor's details, checking if they are saved as a recipient, and then manually entering their routing numbers into a separate system. This process is slow and prone to human error.

Now, your agent takes over. It checks list_recipients for existing payees, or uses create_recipient if you need to add someone new. You just tell it the vendor name, and it executes the full payment setup lifecycle.

Mercury: 10 Tools for Financial Data Management

Use these tools to analyze account balances, review transaction history, list payees, and retrieve core financial records from Mercury.

#	TOOL	DESCRIPTION
01	`list_accounts`	Retrieves a list of every Mercury bank account associated with the business.
02	`list_team_members`	Lists all team members connected to the company's operational accounts.
03	`get_account`	Fetches specific metadata and details for a single bank account.
04	`list_transactions`	Retrieves the most recent transactions, allowing filtering by parameters like date or amount.
05	`get_transaction`	Gets detailed information about a specific transaction ID.
06	`list_recipients`	Shows all payment recipients that have been saved for future use.
07	`create_recipient`	Adds and saves a brand new recipient needed before any payments can be sent out.
08	`list_payments`	Lists payment transactions that have been successfully sent from the account.
09	`list_statements`	Retrieves a list of available bank statements for historical review and auditing.
10	`get_treasury_balance`	Provides an overview or summary of the overall treasury balance across linked accounts.

See It in Action

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

U What's the current balance on all my Mercury accounts?

I queried your Mercury accounts. You have two active accounts: 'Operating Account' with $142,350.00 and 'Savings Reserve' with $500,000.00. Total cash position: $642,350.00.

U Show me all transactions above $5,000 from this month.

I found 4 transactions above $5,000 this month: AWS Infrastructure ($12,400), Payroll Run ($87,500), Office Lease ($8,200), and Legal Retainer ($15,000). Total: $123,100.

U List all my payment recipients.

You have 12 saved recipients. Here are the top 5 by frequency: AWS (ACH), Gusto Payroll (ACH), WeWork (Wire), Stripe Atlas (ACH), and Latham & Watkins (Wire).

Frequently Asked Questions

01 **Can Mercury MCP only read data, or can I send money?**

You can do both. This MCP allows your agent to list payments using list_payments, but it also lets you execute payment actions by listing and creating recipients.

02 **Which tool should I use for a general balance check with Mercury MCP?**

Use get_treasury_balance or list_accounts. These tools give you an immediate, comprehensive overview of the total cash available across all your linked accounts.

03 **How do I find out what transactions occurred last quarter using Mercury MCP?**

You use list_transactions and provide date filters in your prompt. The agent pulls only those records, saving you from scrolling through pages of data.

04 **Is there a way to get account metadata besides the balance via Mercury MCP?**

Yes, the get_account tool specifically retrieves crucial details like routing numbers and general bank account metadata that your code might need.

05 **Do I have to manually update my API key every time I use Mercury MCP?**

No. Once you subscribe via Vinkius, the connection is managed securely through an established token process, allowing continuous data access without constant manual intervention.

06 **How do I get my Mercury API token?**

Log in to your Mercury account at app.mercury.com. Go to **Settings → API Tokens** and click **Create Token**. Choose the token type (Read Only for balance checks, or Read and Write for managing recipients). Copy the token immediately — it will only be shown once. Paste it into the configuration field below, and you're ready. No code, no SDK, no webhooks.

07 **Can my AI agent check my startup's runway instantly during a board meeting?**

Absolutely. Just ask your AI agent 'What's my current balance across all Mercury accounts?' and it returns real-time balances for every account — checking, savings, and treasury — in seconds. No tab-switching, no logging in, no waiting for the dashboard to load.

08 **What if I need to review recent transactions for month-end close?**

Ask your AI agent to pull transactions from any date range. It retrieves each transaction with amount, counterparty, status, and date — perfect for reconciliation. You can filter by account, ask for Wire vs ACH breakdowns, or request only transactions above a specific threshold, all without exporting CSVs.

09 **Is this suitable for companies with multiple Mercury accounts?**

Yes. The integration retrieves data across all accounts linked to your API token — operating accounts, savings accounts, and treasury balances. Perfect for venture-backed startups managing multiple entities, holding companies, or finance teams tracking cash positions across subsidiaries.

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 `"mercury": { "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

Mercury 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 Mercury. 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	Mercury MCP
Server ID	`019d75d2-cfd6-7301-8a79-137c873fa9dd`
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/mercury.