

MCP SERVER

NO CODE

CLOUD HOSTED

OnPay MCP

Manage payroll data and HR records instantly.

OnPay connects your AI client directly to payroll and HR data, letting you programmatically manage everything from employee profiles and payment history to pay schedules and subscriptions. Instead of jumping between multiple internal systems, your agent can pull specific company details or list all current employees in a single conversation.

A+ Quality Score 100/100

payroll

employee-records

tax-filing

benefits-administration

transaction-monitoring

hr-automation



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.

OnPay MCP

8 tools available

Cloud-hosted on Vinkius

Need to audit employee records or run payroll calculations? OnPay gives your AI client direct access to the core data that runs your business. You get programmatic visibility into everything—employee profiles, historical transactions, and future pay cycles. Your agent can instantly list every active employee or retrieve detailed information about a specific payment transaction without needing manual API calls or dashboard navigation.

This means you stop spending time cross-referencing sheets and start making decisions based on real data. By connecting OnPay via Vinkius, your AI client gains the capability to orchestrate complex HR workflows—from viewing active subscriptions to checking company financial item codes for reconciliation. It's simply giving your agent a full view of your payroll operations.

Core Capabilities

01 — Gathering core company details

The MCP can retrieve general information about the associated business entity.

02 — Accessing employee records

It provides detailed profiles for individual employees and can list every person currently registered in the system.

03 — Monitoring payment history

You can pull historical data on payments, including status updates and transaction details.

04 — Scheduling and payroll oversight

The agent monitors upcoming pay cycles and tracks active subscription records for the company.

05 — Financial code retrieval

It pulls necessary item codes and general company settings needed for accounting reconciliation.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to the OnPay MCP on Vinkius.
- 02 Provide your unique OnPay API Token from your management panel.
- 03 Your AI client can now access and manage payroll operations through natural conversation.

The bottom line is, you get to use complex HR data—like listing payment transactions or checking pay schedules—without writing a single line of code.

Built For

This MCP is for the Payroll Specialist who spends too much time manually cross-checking employee records against transaction logs. It's also ideal for Operations Directors who need instant visibility into subscription status or HR Managers running quarterly audits.

Payroll Administrator

They use this to list all employees and pull payment transactions to verify that paychecks matched the correct schedules.

HR Manager

They rely on it to get employee details quickly or check subscription status when onboarding new staff members.

Financial Analyst

They use the accounting integration tools to retrieve item codes and company settings for financial reporting.

What Changes When You Connect

- 01 Stop juggling tabs. Instead of manually listing employees in a separate dashboard, you simply tell your agent to call `list_employees`, and the full list appears immediately for review.

-
- 02** Never lose track of payments again. You can ask the MCP to fetch details on any payment using `get_transaction` or pull an entire history with `list_transactions`. It's instant audit capability.
-
- 03** Stay ahead of pay dates. Use `list_pay_schedules` to monitor upcoming cycles and ensure your team knows exactly when payroll is running, preventing delays and confusion.
-
- 04** Financial accuracy matters. When reconciling accounts, you don't have to guess item codes; the MCP pulls them for you using `list_item_codes`.
-
- 05** Keep track of revenue streams by calling `list_subscriptions`, which immediately shows whether a client is active or if their service ended.
-

Real-World Applications

Running a quick audit on employee status.

The HR Manager needs to verify who has been hired and paid in the last quarter. They ask their agent, 'Show me all active employees.' The MCP uses `'list_employees'` to generate the current roster, saving hours of spreadsheet filtering.

Preparing for tax filing.

The Financial Analyst needs a list of all service subscriptions to calculate accrued revenue. They prompt their agent, 'List all current subscriptions.' The MCP uses `'list_subscriptions'` and provides the complete record set.

Investigating a payment discrepancy.

A team member reports an incorrect payout amount. Instead of calling customer support, the Payroll Admin asks the agent to use `'get_transaction'` for that specific date and ID. They get the precise details needed to resolve the issue instantly.

Planning future payroll runs.

The Operations Director wants to know when the next three pay cycles are scheduled. Asking the agent to use `'list_pay_schedules'` gives them a clear timeline, allowing them to pre-warn department heads.

Patterns to Avoid

Searching by general date range.

X AVOID

Trying to find out if a payment went through last week by manually filtering the web dashboard for 'Last 7 Days'. This often misses records or requires multiple clicks across different views.

✓ INSTEAD

Just ask your agent to run `list_transactions` and filter by date, or use `get_transaction` with the specific ID. It's faster and more reliable.

Guessing accounting codes.

X AVOID

A finance team member needs a code for 'Software Licensing Fees' but isn't sure if it's Code 401 or 410, leading to delays while they email the accounting department.

✓ INSTEAD

Run `list_item_codes`. The MCP gives you the full list of valid item codes immediately so you can use the correct one.

Asking for 'staff data' vaguely.

X AVOID

Saying, 'Can I see the staff information?' This is too vague; it might get them employee names but miss pay status or company details entirely.

✓ INSTEAD

Be specific. Use `list_employees` to get the roster, and then use `get_employee` with a name and ID for detailed records.

The Right Fit

Use this MCP if your core pain point involves reconciling payroll data: comparing employee details, payment history, and accounting codes. If you need programmatic access to list employees (`list_employees`), check pay schedules (`list_pay_schedules`), or verify transaction statuses (`get_transaction`), OnPay is the right fit.

Don't use this if your problem is general communication (like sending emails) or managing CRM contacts for sales leads. For those tasks, you need a different messaging or marketing tool. Also, don't expect it to handle tax filing itself; while it gives you data via `list_item_codes`, the final calculation and submission must happen in dedicated accounting software.

The Headache of Payroll Audits

Right now, checking if a payment is correct means logging into your HR portal to find employee details. Then, you jump to the finance dashboard and search for the transaction ID. You might need to pull up subscription records from a third system just to verify billing status. It's three different logins, five tabs open, and half an hour of copy-pasting.

With this MCP, those steps disappear. Your agent handles it all in one prompt. Need to know if the employee was paid correctly? You ask for the transaction details, and you get the answer instantly, pulling data from multiple sources like payment history and employee profiles into a single clean output.

OnPay MCP: Get Payroll Data in Minutes

You no longer have to manually cross-reference the payroll schedule with active subscriptions. You just ask your agent for both data sets, and it gives you a single view showing exactly when the payment should occur versus when the service is actually billed.

Payroll operations are now immediate and auditable through conversation. The information you need—whether it's listing employees or checking company info—is right there, ready to use.

OnPay: 8 Tools for HR & Payroll Data

Use these 8 tools to pull specific data points on everything from employee profiles to transaction logs.

#	TOOL	DESCRIPTION
01	<code>get_company_info</code>	Gets general information about the company associated with the account.
02	<code>get_employee</code>	Retrieves specific details for a single employee using their unique ID.
03	<code>get_transaction</code>	Pulls detailed information regarding one specific financial payment transaction.
04	<code>list_employees</code>	Retrieves a full list of every employee record in the system.
05	<code>list_item_codes</code>	Lists all available accounting item codes for financial tracking.
06	<code>list_pay_schedules</code>	Retrieves a list of upcoming and past company payroll schedules.
07	<code>list_subscriptions</code>	Lists all active or expired subscriptions managed by the platform.
08	<code>list_transactions</code>	Provides a comprehensive list of payment transactions that occurred.

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 employees.



Fetching employees... I've found 15 active employees. Here are the first few...

Frequently Asked Questions

01 How do I find out if a payment went through using OnPay MCP?

You can check transaction status by calling ``list_transactions`` for an overview, or use ``get_transaction`` with the specific ID. This gives you immediate confirmation on whether the funds were processed.

02 Can I use OnPay MCP to list all employees?

Yes, simply call the ``list_employees`` tool. It provides a comprehensive roster of everyone recorded in your system for immediate review.

03 Does OnPay MCP help with tax filing?

While this MCP gives you access to financial data via ``list_item_codes``, it is designed for data retrieval and reporting. You still need specialized accounting software for the actual tax submission.

04 What if I need company-wide info? Should I use OnPay MCP?







Yes, running ``get_company_info`` pulls general details about your business entity. This is useful before starting any complex payroll calculation or audit.

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 <code>"onpay": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

OnPay 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 OnPay. 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	OnPay MCP
Server ID	019d75e6-3641-70cd-ae0-b1d7174d9d56
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/onpay.