MCP SERVER | NO CODE | CLOUD HOSTED

Getpaid MCP

Automate billing, refunds, and payments instantly.

Getpaid is an MCP that lets you automate billing, process payments, and handle checkouts directly through your AI agent. Connect it to manage customer records, track invoices, and initiate refunds for SaaS or marketplace operations.

A+ Quality Score 100/100

saas-billing | payment-gateway | checkout-automation | transaction-tracking | invoicing

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

SCAN TO VIEW

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.

Getpaid MCP

12 tools available
Cloud-hosted on Vinkius

Running a subscription service means constantly juggling payment statuses, tracking recurring revenue, and verifying transactions—all things that used to require clicking through multiple dashboards. This MCP connects Getpaid's full capabilities to your AI agent, letting you handle these financial tasks using natural conversation instead of manual clicks. You can ask it to list all recent payments or pull up a customer's complete billing history instantly. It handles everything from creating secure checkout sessions to processing partial refunds with just a simple prompt. By connecting this MCP through Vinkius, your agent gets real-time access to payment data without you ever having to leave your primary workspace. You manage complex financial flows—like listing active subscriptions or checking webhook configurations—all while chatting.

Core Capabilities

01 — Handle payments and refunds

Process full or partial refunds against existing charges, or monitor the status of recent transactions.

02 — Manage customer billing data

Retrieve detailed information on specific customers, list all associated invoices, and review their payment history.

03 — Automate revenue collection

Initiate new checkout sessions for products or services and check the status of those payments.

04 — Audit billing activity

List all processed transactions, review recurring subscription plans, and monitor webhook setups.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on the Vinkius Marketplace using your Getpaid Client ID and Secret.

02 Connect it within any compatible AI client (Claude, Cursor, Windsurf, etc.) so your agent can access payment data.

03 Tell your agent what you need—for example, 'List all payments processed last week'—and the system executes the necessary function.

The bottom line is getting complex financial data and actions done through simple conversation rather than navigating a dashboard.

Built For

This connector is built for SaaS founders, Ops Managers, and support teams who spend too much time clicking through dashboards to answer basic billing questions. If your job involves verifying payment status or processing refunds, this saves you hours of manual work.

Customer Support Agent

Uses the MCP to immediately verify if a customer's payment was successful and retrieves specific invoice details without transferring the user to billing.

Operations Manager

Runs reports on revenue flow by listing all payments or reviewing active subscription plans to quickly spot anomalies.

Founding Engineer/Product Owner

Tests payment logic by initiating new checkout sessions and confirming the associated payment links programmatically.

What Changes When You Connect

01 Process refunds without leaving your chat. You can use the refund_billing_payment tool to immediately issue credits for failed or disputed transactions.

02 Stop manually checking statuses. Your agent pulls up payment details using get_payment_details so you know exactly where a customer's money is at any time.

03 Automate setup and testing by running create_checkout_session, which instantly generates secure links for your team to use with clients.

04 Get a complete financial picture. You can list billing invoices or check out list_billing_payments to audit revenue flow across different months.

05 Manage recurring revenue easily. The list_billing_subscriptions tool lets you see if your core subscription plans are active and generating expected income.

Real-World Applications

A customer claims they were double-billed for a service renewal.

Instead of pulling up the dashboard, the agent uses get_payment_details to find two recent charges. You confirm the mistake and use refund_billing_payment to issue the credit immediately, providing proof straight from chat.

You need to onboard a new client and process their first payment.

The agent uses create_checkout_session for the initial amount. Once complete, it checks get_checkout_status to confirm success before sending the welcome materials. This entire flow takes minutes.

An internal audit requires a list of all payment methods used this quarter.

The agent uses list_billing_payments and then filters by date range, compiling a comprehensive report on transaction volume and types without needing access to the raw database UI.

Your team needs to check if webhooks are correctly firing for payment updates.

The agent runs list_payment_webhooks. This quickly confirms that your internal system is configured to receive real-time data notifications, saving manual debugging time.

Patterns to Avoid

Treating the MCP like a database query tool

✗ AVOID

Asking the agent to 'List all payment records and filter by status AND customer name.' This is too complex for a single command.

✓ INSTEAD

Break it down: First, use list_billing_payments to get the ID. Then, pass that ID to get_payment_details, or use get_customer_details first to narrow the scope.

Assuming real-time data visibility

✗ AVOID

Asking the agent about a payment that was processed 10 minutes ago and hasn't shown up in the dashboard yet.

✓ INSTEAD

Always verify the status using get_checkout_status or list_billing_payments. The MCP tells you what it can see, which is better than guessing.

Mixing billing data with general account info

✗ AVOID

Asking 'What's my overall business revenue?' This requires high-level accounting logic outside the scope of this tool.

✓ INSTEAD

Keep it transactional. Use list_billing_subscriptions to see recurring health, or use list_billing_invoices for structured billing reports.

The Right Fit

Use Getpaid if your primary pain point is accessing and acting upon real-time financial transaction data. Specifically, you need to create payment links, verify customer payments, process refunds, or audit subscription plans. Don't use this MCP if you are building a full GAAP ledger system or require tax calculation based on local jurisdiction laws; it handles the *transactions*, not the final accounting reconciliation. If your goal is simply general data retrieval without any action (like 'Show me all user emails'), then a basic contact management tool would be better. But if that data needs to trigger an action—like refunding money or generating an invoice—this MCP is exactly what you need.

Handling Payments Used To Be A Mess of Tabs and Spreadsheets.

Today, checking a single payment status means opening the dashboard, finding the customer ID, navigating to the payments tab, searching by date range, filtering by status, and then manually copying that information into an email or spreadsheet for your manager. If you have five customers asking about their billing history, it's a multi-tab ordeal.

With this MCP, you just ask your agent: 'What is the payment status for customer X?' It executes get_payment_details and spits out the clean answer right in the chat. The manual clicking stops; the instant data access begins.

Getpaid Powers Billing Operations with One Conversation.

The need to manually generate payment links, process refunds via a separate portal, and reconcile subscription status across different systems disappears. Your agent handles

Your entire financial workflow moves from multiple web pages back into the chat window. You don't just view data; you act on it.

create_checkout_session for new sales and uses refund_billing_payment when things go wrong.

Getpaid: 12 Billing & Payment Tools

Use these functions to perform critical financial tasks like creating checkouts, listing invoices, and processing refunds directly through natural language commands.

#	TOOL	DESCRIPTION
01	`verify_api_connection`	Checks if your AI agent has successfully connected to your Getpaid account.
02	`create_checkout_session`	Starts a new payment session for a specific amount, generating a unique checkout link.
03	`get_account_details`	Retrieves general information about your Getpaid account identity.
04	`get_checkout_status`	Checks the current status of a payment session using its unique ID.
05	`get_customer_details`	Fetches detailed metadata for any customer associated with your billing account.
06	`get_payment_details`	Retrieves full payment metadata, including status and timestamps, for a specific transaction.
07	`list_billing_customers`	Lists all customers who have paid through your platform.
08	`list_billing_invoices`	Retrieves a list of all generated invoices and their corresponding details.
09	`list_billing_payments`	Lists every processed payment record, allowing you to monitor your overall revenue flow.
10	`list_billing_subscriptions`	Shows all active and inactive recurring billing plans for subscription management.
11	`list_payment_webhooks`	Displays the configurations of webhooks to confirm real-time notification setup.
12	`refund_billing_payment`	Processes a full or partial refund for any captured payment directly through the agent conversation.

See It in Action

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

U List all payments processed today.

Retrieving payments... I found 5 payments processed today, including a $50.00 charge (Status: Captured) and a $15.00 charge. Would you like the full details for any of them?

U Create a checkout session for $25.00 USD.

Checkout initiated! I've successfully created a session for 2500 cents ($25.00 USD). The Checkout ID is 'chk_abc123'. You can share this link with the customer: [Payment Link].

U Refund payment 'pay_98765' for the full amount.

Refund processed! I have successfully initiated a full refund for payment pay_98765. The status is now 'refunded' and the customer will see the credit shortly.

Frequently Asked Questions

01 **How do I list all payments using Getpaid MCP?**

You use the list_billing_payments tool. This function gathers every payment record, giving you a clear overview of your recent revenue flow directly within the chat.

02 **Can I process refunds with Getpaid MCP?**

Yes. You call refund_billing_payment and provide the specific payment ID to initiate the full or partial credit immediately through your agent.

03 **What is the difference between list_billing_payments and list_billing_invoices?**

list_billing_payments shows every transaction (the actual money movement). list_billing_invoices provides the structured records of what was billed to the customer.

04 **Does Getpaid MCP help with new checkouts?**

Absolutely. You use create_checkout_session to initiate a secure payment flow, which gives you a unique link and status that your agent can track for you.

05 **Is get_customer_details enough for billing needs?**

No. While it gets basic customer metadata, use list_billing_customers or get_payment_details to get the actual financial records needed for reporting.

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 `"getpaid": { "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

Getpaid 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 Getpaid. 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	Getpaid MCP
Server ID	`019d75a3-c652-7163-9634-38063f2c0e93`
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/getpaid.