MCP SERVER | NO CODE | CLOUD HOSTED

MarketMan MCP

Manage Inventory, Orders & Waste in Conversation

MarketMan MCP manages every aspect of restaurant operations through conversation. It handles real-time inventory tracking, drafts purchase orders for vendors, analyzes recipes to calculate food costs, and monitors spoilage waste—all without needing to click tabs or copy spreadsheets.

A+ Quality Score 100/100

restaurant-management | food-cost-analysis | purchase-orders | waste-tracking | supply-chain-visibility

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

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.

MarketMan MCP

8 tools available
Cloud-hosted on Vinkius

Need to know if you can actually open for service? This MCP connects your agent directly to the core intelligence platform that runs a restaurant. You talk to it naturally, and it pulls data on everything from stock levels to supplier history. It doesn't just tell you what you need; it helps you draft the actual purchase orders or identify recipes that are costing too much. The entire catalog of tools is managed via Vinkius, meaning your agent can access this restaurant intelligence alongside dozens of other systems you use daily.

Whether you're checking current stock levels against par targets, reviewing supplier details, or generating a count sheet for the day, it all happens in conversation. You get accurate data on waste events and can analyze food costs per plate instantly.

Core Capabilities

01 — Check Current Stock Levels

List every inventory item to see what's low or needs reordering.

02 — Manage Supplier Information

Access and organize details about all your vendors and their order history.

03 — Generate Procurement Lists

Create a list of current purchase orders, tracking deliveries and invoices for review.

04 — Analyze Menu Costs

List all existing recipes to analyze ingredient quantities and associated food costs.

05 — Track Spoilage

Review past waste events to identify patterns in spoilage and reduce losses.

06 — Audit Stock Counts

Pull up current inventory count sheets for detailed variance tracking during audits.

One Click on Vinkius — From Prompt to Execution

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

01 Tell your agent what data you need—for example, 'What's our stock level on beef?' or 'List all vendors.'

02 The MCP runs the appropriate tool against MarketMan and pulls the structured data (e.g., low inventory items, vendor lists, waste logs).

03 Your agent processes that raw data, presenting you with an immediate, conversational summary or a draft document ready for review.

The bottom line is you get real-time operational insights without ever leaving your chat interface.

Built For

This MCP is built for restaurant operations managers, head chefs, and inventory controllers. It targets the constant stress of balancing supply chain demands with fluctuating menu costs.

Head Chef

Uses this to quickly check ingredient quantities needed for a new recipe or run cost analysis on existing dishes.

Operations Manager

Checks inventory levels across different categories, reviews purchase orders, and monitors waste events to keep the store running smoothly.

Inventory Controller

Generates count sheets for physical audits and retrieves detailed vendor information to reconcile discrepancies.

What Changes When You Connect

01 Stop manually checking stock sheets. You ask about low items, and the agent immediately lists everything below par level using list_inventory.

02 Drafting orders is fast. Instead of logging into a separate system to see what needs ordering, you just talk to your agent and it accesses purchase order data via list_purchase_orders.

03 You can nail down food cost percentages quickly. Use the list_recipes tool to pull up all ingredients and analyze their quantities for menu tweaks.

04 Vendor management is cleaner. You don't need to open multiple tabs; you simply ask for a vendor list using list_vendors and get order history right there.

05 Waste tracking becomes proactive. The agent uses get_waste_events to show you where spoilage happens, helping you adjust your buying habits immediately.

06 Auditing is simplified. Instead of printing physical sheets, the agent pulls up current count sheets via get_count_sheets for instant variance checking.

Real-World Applications

Running End-of-Day Inventory Checks

The manager asks their agent, 'What's our inventory status?' The agent uses list_inventory to identify three items below par levels and simultaneously runs get_count_sheets if the count was done manually. This prevents running out of key ingredients before morning service.

Revising a High-Cost Menu Item

The head chef asks, 'What's the cost breakdown for the Signature Burger?' The agent uses list_recipes to retrieve the full ingredient data and calculate the current food cost percentage, flagging any expensive components.

Handling a Supplier Dispute

The operations manager asks about a specific vendor. The agent uses list_vendors to pull up the supplier's full order history, allowing them to quickly prove discrepancies or track overdue deliveries.

Planning for Peak Season Ordering

Instead of manually compiling lists, the manager asks to draft an order. The agent uses list_purchase_orders and compiles a draft request including needed items and expected costs from Fresh Farms Inc.

Patterns to Avoid

Mixing up systems

✗ AVOID

Trying to calculate food cost by cross-referencing spreadsheet data with vendor invoices. This is time-consuming, prone to human error, and involves copy/pasting dozens of numbers.

✓ INSTEAD

Just ask your agent to list_recipes. It handles the ingredient quantities and calculates the food cost directly from MarketMan's source data.

Overlooking waste

✗ AVOID

Assuming spoilage is normal overhead because you don't have a centralized way to track it, leading to unnecessary losses.

✓ INSTEAD

Run get_waste_events. The system shows exactly what spoiled and its associated cost impact, helping you reduce future loss.

Manual stock checks

✗ AVOID

Wasting time manually counting every item in the back room and then having to input those numbers into a separate spreadsheet just to check against par levels.

✓ INSTEAD

Ask the agent to get_count_sheets. It pulls up the necessary count sheets, so you only need to focus on the physical count.

The Right Fit

Use this MCP if your biggest bottleneck is accessing and interpreting complex operational data across multiple departments—inventory, purchasing, recipes, and waste. If you frequently find yourself switching between a POS system, an inventory spreadsheet, and a separate accounting tool just to answer the question, 'Can we afford this menu change?', then this is for you. You need a single conversation point that pulls all those disparate systems together.

Don't use this if your primary problem is simple data entry or basic record keeping. If you only need to create a simple list of names, or send a standalone message, an existing messaging tool will work fine. This MCP excels at deep analysis and cross-domain comparison; it doesn't just list things, it analyzes the relationship between inventory status, recipe costs, and vendor reliability.

The struggle to keep accurate counts across multiple systems is draining.

Right now, tracking what you have versus what you need requires a painful dance. You open the spreadsheet for current stock levels, then jump over to the purchasing portal to see where your last order was delivered, and finally pull up an old invoice just to verify pricing. It's all clicking through tabs, cross-referencing dates, and copy/pasting numbers until you feel like you need a vacation.

With this MCP, that entire process disappears. You simply ask your agent what the inventory status is for key items. The system automatically checks current stock levels and compares them to par targets, giving you one clear answer without opening a single new tab.

MarketMan MCP gives you full visibility into ingredient costs.

Previously, determining the food cost for a dish was guesswork. You'd manually gather recipe cards, estimate portion sizes, and then cross-reference current wholesale prices from your vendors—a process that took hours and inevitably led to outdated costing.

Now, you ask the agent about a specific menu item. It uses list_recipes to pull up every ingredient quantity, calculates the cost using real vendor data, and tells you the exact food cost percentage in seconds.

MarketMan: 8 Tools for Restaurant Operations

These tools let you manage every facet of restaurant operations, from tracking low stock to analyzing complex recipes and waste history.

#	TOOL	DESCRIPTION
01	`list_inventory`	Retrieves a list of all current inventory items, helping you check stock levels against par targets.
02	`list_purchase_orders`	Lists existing purchase orders so you can track deliveries and review invoices from suppliers.
03	`list_vendors`	Provides a comprehensive list of all vendors used by the restaurant, including their order history.
04	`list_categories`	Lists inventory categories and shows item counts within each group for high-level tracking.
05	`get_count_sheets`	Generates current inventory count sheets to help you track variance during manual audits.
06	`list_recipes`	Lists all recipes in the system, allowing you to analyze ingredient quantities and food costs for menu engineering.
07	`get_waste_events`	Retrieves records of waste events, helping you identify patterns in spoilage and reduce unnecessary loss.
08	`get_restaurant_info`	Pulls general information about the restaurant location or entity for context.

See It in Action

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

U Are we low on any inventory items?

📦 **Low Inventory Alert**

⚠️ **Below Par Level**:
- 🥩 Ribeye: 8 lbs (par: 20) — **ORDER NOW**
- 🧈 Butter: 5 lbs (par: 15) — order by Friday
- 🍋 Lemons: 12 ct (par: 30) — order by Thursday

✅ Everything else is within range (47/50 items OK)

U Draft a purchase order for our primary produce vendor.

I have prepared a draft purchase order for 'Fresh Farms Inc.' It includes 20 lbs of tomatoes, 15 lbs of onions, and 10 bunches of basil to bring you up to par levels. Total estimated cost is $145. Would you like to review?

U What is the current food cost percentage for our signature burger?

The food cost for the Signature Burger is currently at 28.5%. The recent $0.50 increase in beef prices from your wholesale supplier raised the cost per portion by 3%.

Frequently Asked Questions

01 **How does MarketMan MCP help with inventory counts?**

It simplifies physical audits by generating count sheets. Instead of starting from scratch, you can ask the agent to get_count_sheets for an accurate baseline when performing a manual stock take.

02 **Can I use MarketMan MCP to draft purchase orders?**

Yes, you can. The MCP pulls current order data and allows your agent to help draft the necessary purchase orders based on what needs replenishing or revising.

03 **Is MarketMan MCP better than a standard ERP system?**

It's different. A traditional ERP is where the data lives; this MCP uses natural conversation to access and analyze that data, presenting actionable insights instantly without requiring specialized queries or coding.

04 **What kind of waste tracking does MarketMan MCP provide?**

It gives you detailed records via get_waste_events. You can ask for past spoilage reports to identify patterns and reduce the money you lose to food waste.

05 **Does MarketMan MCP help with menu engineering?**

Absolutely. By listing recipes and analyzing ingredient quantities, it gives chefs the data needed to adjust portion sizes or swap expensive ingredients without impacting taste.

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 `"marketman": { "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

MarketMan 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 MarketMan. 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	MarketMan MCP
Server ID	`019d75cf-48f7-7346-b8e8-0aae137e78e5`
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/marketman.