MCP SERVER | NO CODE | CLOUD HOSTED

GrabFood Partner MCP

Control everything from order acceptance to marketing campaigns.

GrabFood Partner MCP automates your entire restaurant delivery operation directly from your AI agent. Manage incoming orders, update prices instantly, control store visibility, and launch marketing campaigns without ever logging into a separate dashboard again.

A+ Quality Score 98.33/100

delivery-management | restaurant-operations | order-lifecycle | menu-updates | food-delivery

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/grabfood-partner

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.

GrabFood Partner MCP

12 tools available
Cloud-hosted on Vinkius

Connecting your GrabFood account lets you take full control of your restaurant operations using natural conversation with any compatible AI client. Forget jumping between portals during the lunch rush. Your agent handles everything from order intake to menu changes. You can tell it to check if an incoming order is acceptable or even reject it outright, flagging the reason for the record.

It's not just about accepting orders; you control the entire lifecycle. Tell your AI client to run a batch update on your menu prices or mark several items as sold out immediately. Need to pause the store because the kitchen is overwhelmed? You can do that too. The ability to manage these complex, multi-step tasks makes this MCP an essential piece of infrastructure. By connecting through Vinkius, you get one catalog access point that lets your agent handle all facets of your delivery business.

Core Capabilities

01 — Manage incoming orders

Accept, reject, or mark received orders as ready for pickup using simple commands.

02 — Update menu inventory and pricing

Make targeted changes to your offerings by updating specific item prices or marking entire groups of items unavailable.

03 — Control store operational status

Instantly pause, unpause, or check the current open/closed status of your physical location on the platform.

04 — Plan and launch promotions

Create new marketing campaigns, like discounts or free delivery offers, to boost visibility and orders.

05 — Manage order timing estimates

Adjust the estimated time an order will be ready for pickup if kitchen prep falls behind schedule.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and connect your GrabFood Partner OAuth2 access token.

02 Next, tell your AI client what you need. For example: "Check if my store is open." or "Batch-update the prices for these three dishes."

03 The agent sends the request to the platform, executes the action (like accepting an order or pausing the store), and returns a confirmation of the status change.

The bottom line is you get real-time operational control over your entire restaurant flow without manual dashboard interaction.

Built For

This MCP is for the Restaurant Owner who's tired of clicking through multiple dashboards during a lunch rush, or the Operations Manager needing to monitor several locations simultaneously. If your job involves reacting quickly to order status changes and inventory issues, this tool saves hours of manual dashboard navigation.

Restaurant Owner

Accepting orders, updating basic menu items, and managing the store's open/closed status from a tablet while in the kitchen.

Operations Manager

Monitoring multiple locations for optimal availability; running batch updates to mark temporary outages or adjusting estimated ready times across different shifts.

Marketing Coordinator

Launching and tracking promotional campaigns, reviewing past campaign performance, and ensuring the store is visible when needed.

What Changes When You Connect

01 Stop logging into the portal during peak hours. Your AI client handles accepting and rejecting incoming orders instantly, giving you uninterrupted focus on the kitchen.

02 Never worry about outdated pricing again. Use `batch_update_menu` or `update_menu` to change prices or mark items as unavailable across dozens of dishes in seconds.

03 Instantly control visibility by using `pause_store`. Need a quick break? Pause your store with a single prompt, letting you reset without losing track of the platform.

04 Boost sales with precision. Use `create_campaign` to launch time-limited discounts or free delivery offers exactly when you need them most.

05 Keep drivers happy and avoid delays. When prep takes longer than expected, use `update_ready_time` to proactively manage expectations without manual calls.

Real-World Applications

Handling the Lunch Rush Spike

The restaurant owner sees a sudden spike of 20 new orders. Instead of manually clicking 'Accept' on each one, they tell their agent to process all pending orders and confirm readiness for the first batch using `list_orders`, followed by accepting them via `accept_order`.

Inventory Crisis Management

The Operations Manager realizes three signature dishes are out of stock. They don't want to update the whole menu, so they use `batch_update_menu` to mark only those specific items as unavailable immediately.

Seasonal Menu Overhaul

The Marketing Coordinator is launching a summer promotion and needs to add 15 new seasonal dishes. They use the comprehensive `update_menu` tool, replacing the old menu structure with the full new catalog data.

Mid-Day Operational Break

The Manager notices staff are exhausted and wants a temporary break without losing visibility. They check their status using `get_store_status` and then tell the agent to `pause_store`, stopping new orders immediately.

Patterns to Avoid

Massive Menu Replacement

✗ AVOID

Thinking you have to upload the entire menu file every time a price changes, leading to hours of data preparation and risk.

✓ INSTEAD

Don't use `update_menu` unless absolutely necessary. For small changes, always prefer using `batch_update_menu` or targeted updates to only the specific item identifiers that need modification.

Ignoring Order Status

✗ AVOID

Accepting an order when staff know they are currently overwhelmed and might not be able to fulfill it, damaging performance metrics.

✓ INSTEAD

Always check first. Before accepting or canceling, use `check_order_cancelable` to verify the current state of the order, ensuring you act within the permitted window.

Over-reliance on Full Status Checks

✗ AVOID

Asking your agent to run `get_store_status` repeatedly just to confirm if the store is open.

✓ INSTEAD

If you suspect a status issue, first try using `pause_store` or `update_ready_time`. If those actions fail or revert immediately, *then* use `get_store_status` to diagnose why.

The Right Fit

Use this MCP if your daily job involves constant decision-making around the order lifecycle: accepting, rejecting, modifying inventory, pausing operations, or running promotions. It's perfect for operational staff who need a single source of truth and action capability.

Don't use this if you only want to read data (e.g., just listing orders). For pure retrieval, simple API wrappers might suffice. But if you need the system to *act*—if you need it to change prices, toggle store status, or mark something as ready—this MCP is required. If your goal is complex reporting across multiple platforms, look into a data warehousing connector instead.

Dealing with Order Chaos and Menu Updates

Right now, managing an unexpected spike in orders means jumping between the GrabFood portal and your POS system. You manually click through incoming notifications, verify each order ID against inventory sheets, then stop to adjust prices because a key ingredient ran out. It's constant context switching, copy-pasting IDs into spreadsheets, and hoping you didn't miss anything.

With this MCP, the process changes entirely. Instead of clicking, you tell your agent what needs doing. You ask it to check incoming orders using `list_orders`, then instruct it to accept or reject batches, all while simultaneously telling it to run a targeted price change using `batch_update_menu`. The result is immediate action without leaving the chat window.

GrabFood Partner MCP: Control Order Acceptance and Menu Updates

Specific manual steps that disappear include logging in to confirm operational status, manually setting new prices for specific dishes, or physically marking an order as ready after the

Now, you've got immediate, conversational control. You dictate the actions—from checking if cancellation is possible with `check_order_cancelable` to updating the entire

kitchen finishes prep. All of this was once a tedious sequence of clicks across different tabs.

menu structure via `update_menu`. It's seamless command execution.

GrabFood Partner: 12 Tools for Restaurant Operations

Use these tools to control every aspect of your GrabFood business. Accept orders, update menus in batches, manage store visibility, and run marketing campaigns using natural language commands.

#	TOOL	DESCRIPTION
01	accept_order	Accepts or rejects a specific GrabFood order ID, committing the restaurant to prepare it or notifying the customer of rejection.
02	batch_update_menu	Updates prices, availability, and descriptions for multiple selected menu items at once.
03	cancel_order	Cancels an already accepted order, requiring a specific reason that impacts the merchant's performance score.
04	check_order_cancelable	Verifies if an incoming order can still be cancelled by checking its current status in the fulfillment pipeline.
05	create_campaign	Creates a new promotional campaign, defining details like discounts or free delivery offers for increased visibility.
06	get_store_status	Retrieves the current operational state of the store, showing if it's open, paused, or closed to customers.
07	list_campaigns	Displays all active and past marketing campaigns associated with the merchant ID for review.
08	list_orders	Fetches a list of incoming orders for a specific date, including details about items and customer information.
09	mark_order_ready	Notifies the platform that an order has been fully prepared and is ready for the driver to pick up immediately.
10	pause_store	Temporarily suspends receiving new orders, making the store invisible until it's unpaused again.
11	update_menu	Replaces or modifies the entire menu structure, including categories and item details, requiring a complete data payload.
12	update_ready_time	Adjusts the estimated time the order will be ready for collection to manage driver dispatch timing accurately.

See It in Action

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

U Show me all my GrabFood orders for today.

I've fetched your GrabFood orders for today. You currently have 8 active orders, with 3 pending acceptance and 5 being prepared. The most recent order (#GF-48921) includes 2x Nasi Goreng and 1x Es Teh Manis. Would you like to accept all pending orders?

U Accept order GF-48921 and mark order GF-48915 as ready.

Done! Order GF-48921 has been accepted and the kitchen can now start preparing it. Order GF-48915 has been marked as ready for pickup — the Grab driver will be notified to arrive shortly.

U Check my store status and pause it if it's currently open.

Your GrabFood store is currently **open** and accepting orders. I've paused it for you — the store is now hidden from customers and no new orders will come through. Existing active orders are unaffected.

U Update the price of my Nasi Goreng to 25000 IDR and mark Fried Rice as unavailable.

I've batch-updated your menu: Nasi Goreng price is now 25,000 IDR and Fried Rice has been marked as unavailable. Customers will see these changes reflected on the GrabFood app within a few minutes.

Frequently Asked Questions

01 **How do I get my GrabFood Merchant ID?**

Your Merchant ID is provided when you register as a GrabFood Partner. You can find it in the GrabFood Partner Portal under your restaurant settings, or in the welcome email you received when your restaurant was onboarded to GrabFood.

02 **Can I use this integration in production, not just sandbox?**

Yes! The integration defaults to the GrabFood sandbox environment for safe testing. To switch to production, you'll need to update the base URL in the engine configuration to the production endpoint. Contact GrabFood Partner support for production API credentials.

03 **What happens if I reject an order? Does it affect my merchant rating?**

Rejecting orders may impact your merchant performance metrics on GrabFood. Frequent rejections can lead to lower visibility in search results. It's recommended to only reject orders when absolutely necessary (e.g., item out of stock, kitchen closure). Always verify order details before making a decision.

04 **Can I update menu items individually without replacing the entire menu?**

Yes! Use the `batch_update_menu` tool for targeted updates to specific items, prices, or availability. This is safer than a full menu replacement since it only touches the items you specify. Use `update_menu` only when you need to replace the entire menu structure.

05 **How do I get my OAuth2 access token for the GrabFood Partner API?**

You can obtain your OAuth2 access token through the GrabFood Partner Portal. Log in to your partner account and navigate to API Settings or Developer section to generate or retrieve your token. If you don't have access, contact your GrabFood Partner account manager.

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 `"grabfood-partner": { "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

GrabFood Partner 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 →

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by GrabFood Partner. 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	GrabFood Partner MCP
Server ID	`019d75aa-58c3-713e-a23a-596c17f5e6a1`
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/grabfood-partner.