

MCP SERVER

NO CODE

CLOUD HOSTED

Booqable MCP for AI Agents

Manage rental inventory and track product availability instantly

Booqable MCP manages your entire rental life cycle directly through natural conversation. Use this connection to track inventory levels, check real-time availability for specific items, and manage orders from concept all the way through pickup and return.

A+ Quality Score 100/100

rental-software

order-lifecycle

inventory-tracking

stock-management

business-operations

catalog-management



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.

Booqable MCP

10 tools available

Cloud-hosted on Vinkius

Manage your physical rentals—from tracking initial customer interest to marking final item returns—without opening a dashboard. This MCP connects Booqable's core inventory and order systems directly to your AI client, letting you orchestrate complex workflows using plain language.

For example, instead of running three separate reports (one for customers, one for stock, and one for orders), you can ask your agent to 'List all open rentals that need pickup today.'

The MCP handles the lookups and combines the results instantly. You get a single, actionable answer. If your existing workflows are complex, connecting through a catalog like Vinkius makes it easy for any compatible AI client—whether Claude or Cursor—to access this data.

The system lets you view product group catalogs, manage customer profiles, check specific dates for item availability to prevent double-bookings, and move orders from concept status all the way to marked as picked up or returned. It puts complete operational control into your agent's hands.

Core Capabilities

01 – Checking Item Availability

You can verify if a specific product group has enough units available for any set of dates.

03 – Viewing and Creating Orders

The agent can retrieve full details for specific orders, list all current rentals, or even create a new order concept.

05 – Controlling Order Lifecycle

Directly change an order's status, marking it as picked up (started) or marked as returned (stopped).

02 – Managing Customer Records

Retrieve detailed information on existing customers or list every customer in your system.

04 – Monitoring Inventory Status

You can access your product groups and technical catalog details to ensure the inventory data is accurate.

One Click on Vinkius — From Prompt to Execution

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

- 01 First, subscribe to this MCP and provide your Booqable API Key and Subdomain credentials.
- 02 Next, connect the MCP to your preferred AI client—like Cursor or Claude—through Vinkius.
- 03 Finally, simply talk to your agent. Ask it anything about orders, availability, or customers, and it executes the commands instantly.

The bottom line is, you bypass the dashboard entirely. You just ask your AI client a question, and it uses Booqable's data directly.

Built For

This MCP is built for anyone running a physical rental business who hates clicking through multiple tabs to get one simple answer. If you manage inventory or orders across different locations, this is your fix.

Operations Lead

Monitoring stock levels and checking order status for pickups and returns without having to open the main dashboard.

Support Specialist

Verifying a customer's account details or checking if specific items are available for their requested dates while chatting with them.

Business Owner

Getting an immediate overview of the entire order pipeline, from concepts to completed returns, all in one chat window.

What Changes When You Connect

- 01 Stop jumping between tabs. You can get a full overview of all active rentals by calling `list_orders`, which shows you where every item stands without leaving your chat window.

-
- 02 Prevent double-bookings before they happen. Use `check_availability` to confirm if an item is free for those specific dates and times, saving manual cross-checking time.

 - 03 Keep your team moving fast. When a pickup happens, just tell your agent to `start_order`, updating the status immediately and accurately in Booqable's system.

 - 04 Never lose track of who rented what. With `get_customer` and `list_customers`, you have instant access to complete client history and contact details right when you need them.

 - 05 Better catalog control means less manual effort. By using `list_product_groups`, your agent can pull up technical specifications for any item group instantly.
-

Real-World Applications

Finding items for a new client booking

A support specialist needs to know if the 'Camping Chair' kit is available from September 1st to September 5th. They ask their agent, and it uses `check_availability` immediately, confirming stock without them needing to navigate complex date calendars.

Checking on overdue returns

A manager needs to know if Order #402 is still out. They use `get_order`, and the agent pulls up the full status and due date in seconds, eliminating manual follow-up work.

Closing out a day's pickups

The operations lead finished a batch of rentals. Instead of opening the dashboard and changing statuses one by one, they ask the agent to `start_order` for all 15 pending orders, marking them picked up instantly.

Building a new rental package

A sales rep needs to see what product groups are available for booking. They ask the agent to `list_product_groups`, getting a full catalog view that they can then use to recommend items.

Patterns to Avoid

Trying to update statuses manually

✗ AVOID

A user opens the Booqable dashboard and clicks through 20 different orders, changing their status one by one in a tedious process.

✓ INSTEAD

Instead, use your AI client to send a single command: 'Start all pending pickups for today.' This triggers `start_order` across multiple records instantly.

Using old customer data

✗ AVOID

A support agent has the wrong phone number for a client and calls them, wasting time.

✓ INSTEAD

First, use `list_customers` to find the correct profile ID, then use `get_customer` with that ID. This ensures your AI agent pulls the latest, verified contact information.

Assuming availability

✗ AVOID

A sales rep promises a client an item is free for two weeks, only to find out later when opening the dashboard that it was double-booked.

✓ INSTEAD

Always verify first. Ask your agent to run `check_availability` before confirming any booking dates. This prevents scheduling conflicts.

The Right Fit

Use this MCP if your pain point is coordinating information spread across multiple Booqable views—specifically, linking a customer's profile (`list_customers`) to their order status (`get_order`) and then checking that specific item's availability (`check_availability`). It's perfect for operational teams. Don't use this MCP if you only need to read static product information; in that case, a simple catalog API might suffice. You also don't need it if your team exclusively manages rentals through a completely separate system. This MCP excels when you need to *act* on the data—creating an order (`create_order`) or changing status (`start_order`)—using natural language commands.

Booqable MCP for AI Agents: Streamlining Rental Order Lifecycle Management

Currently, managing a rental order feels like jumping through hoops. You have to log into Booqable, find the customer's record, manually check the product group catalog for details, and then navigate to the specific order page just to see if it's ready to be marked as picked up or returned. It's slow, click-heavy work that takes time away from actual service.

With this MCP, you simply tell your agent what needs doing. 'What is the status of Order #102 and can we book a replacement item?' The system handles all those internal steps: it pulls the order details, checks the product group inventory, and gives you one clear answer right there in the chat window.

Booqable MCP for AI Agents: Accurate Inventory Tracking and Availability Checks

The biggest pain point is availability. You can't trust memory or a glance at a dashboard; you need to confirm if 'Camera Kit' has enough units for the exact dates requested. This requires cross-referencing product groups against booking schedules—a manual task that invites error.

This MCP automates that entire check. By utilizing the `check_availability` tool, your agent gives you real-time confirmation on unit count and date range instantly, making overbooking a thing of the past.

10 Tools for Booqable: Order Lifecycle and Inventory Control

Use these tools to check item availability, create new rentals, manage customer profiles, and control the status of every order in your system.

#	TOOL	DESCRIPTION
01	<code>check_availability</code>	Checks if a specific item group has enough units available for given dates.
02	<code>create_order</code>	Generates and submits a new rental order concept into the system.
03	<code>get_customer</code>	Retrieves specific details about one customer using their ID or profile information.
04	<code>get_order</code>	Fetches all necessary details for a particular order number.
05	<code>get_product_group</code>	Gets detailed catalog information about an entire group of products, like 'Cameras' or 'Tents'.
06	<code>list_customers</code>	Returns a list of all customers currently registered in the Booqable system.
07	<code>list_orders</code>	Retrieves a comprehensive list of all existing rental orders, showing their current status.
08	<code>list_product_groups</code>	Provides an exhaustive catalog listing of every product group available for rent.
09	<code>start_order</code>	Changes the status of a specific order to 'Started,' marking it as picked up from inventory.
10	<code>stop_order</code>	Changes the status of an order back to 'Stopped,' marking it as returned to usable stock.

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 rental orders for next week and tell me which ones need to be picked up.



Rental Status Report: Next 7 Days

Here are your open rentals:

- **Order #1015:** Status: Reserved. Items: Tent, Sleeping Bag.
- **Order #1022:** Status: Concept. Items: Camera Kit, Drone.
- **Order #1030:** Status: Active (Needs Pickup).

Please note that Order #1030 needs action. Would you like me to check its current inventory status?

- U** Can I book a 'Sound System' from June 25th to the 3rd? Check availability.



Availability Check: Sound System

I checked the catalog for this period. Good news! The 'Sound System' group has **8 units available** for those dates. You can proceed with creating a concept order if you like.

Units Available: 8/12

Dates Checked: June 25th - July 3rd

U Update Order #900: Mark it as returned and tell me the customer's email.



Order Update Complete

✔ Order #900 has been successfully marked as **Stopped (Returned)**. The items are now fully back in your available inventory.

Client Details: John Doe

Contact Email: john.doe@example.com

Frequently Asked Questions

01 How does the Booqable MCP help with day-to-day order status updates?

It lets you update order statuses (like marking an item as returned or picked up) using simple conversation. You don't need to open a dashboard; just tell your agent, and it makes the change immediately.

02 Can I use Booqable MCP to check if enough items are available for multiple dates?

Absolutely. The system runs availability checks across specific product groups using defined date ranges, preventing you from accidentally double-booking inventory.

03 What kind of customer information can the Booqable MCP access?

It gives your agent access to your entire customer directory. You can list all clients or retrieve detailed profiles for specific individuals whenever needed.

04 Is the Booqable MCP good for building new rental packages?

Yes, you can use it to view the full product catalog by listing available product groups. This helps your agent gather all necessary details to recommend and quote entire package deals.

05 Does the Booqable MCP connect my inventory management to general tasks?

Yes, it connects Booqable's core functions—orders, customers, and catalog data—into a single workflow. You can manage complex operations through natural conversation.

06 If I run into issues with an order, how does the MCP help me diagnose it?

You can use the `get_order` tool to pull up every detail of a specific rental, letting you quickly see its history and current status without manually digging through reports.

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 `"booqable": { "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

Booqable 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 Booqable. 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	Booqable MCP
Server ID	019d7561-5a3e-7196-8d25-827b5da97bca
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/booqable.