

MCP SERVER

NO CODE

CLOUD HOSTED

Squarespace MCP

Audit Orders, Inventory, & Customer Data Live

Squarespace MCP connects your AI client directly to your e-commerce store's backend data. It lets you read all your orders, track inventory levels, audit customer profiles, and analyze financial transactions without logging into the platform. You get a single conversational source for every piece of shop data.

A+ Quality Score 100/100

website-management

transaction-auditing

inventory-audit

customer-data

read-only-access

cms-integration



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.

Squarespace MCP

6 tools available

Cloud-hosted on Vinkius

Instead of navigating through multiple dashboards or exporting CSV files just to find out if a specific client paid their balance, you talk to your AI agent. This MCP gives your agent read-only access to everything happening in your Squarespace store—from the moment an order is placed until it ships. Your AI can pull together full customer records by running `list_member_profiles` and then cross-reference those users against recent financial movements using `list_transactions`. You stop hunting for data across different tabs and start asking questions. Vinkius brings all this structured e-commerce intelligence into one context, letting your agent perform complex audits or simply answer, 'Did John Doe pay his balance last week?' instantly.

Core Capabilities

01 — Audit Customer Data

Retrieve and map full customer records by listing profiles for contact details and purchase history.

03 — Assess Stock Levels

Get a real-time count and list of available stock for every product in your catalog.

05 — Examine Product Catalogues

View a complete list of every product available in the store, including general details and current stock counts.

02 — Track Shipment Statuses

List all commerce orders, then pull specific line items to check the status of unfulfilled or processed shipments.

04 — Review Sales History

Pull comprehensive lists of all financial transactions to track revenue, refunds, and payments.

One Click on Vinkius — From Prompt to Execution

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

- 01 Start by activating this read-only MCP module connection within your preferred AI client.
- 02 Securely provide your developer API key to the host. This establishes the pipeline between the AI and your Squarespace store data.
- 03 Send a natural language request—for instance, asking the agent to 'list all products with low stock'—and receive an immediate, compiled answer.

The bottom line is you give your agent permission to read everything, and it uses that access to pull specific data points into one unified response for you.

Built For

Anyone who manages a live e-commerce store through Squarespace. It's perfect for the operations manager tired of switching between the admin dashboard and their spreadsheet, or the founder who needs an instant audit report without writing complex SQL queries.

E-commerce Manager

Uses this MCP to check if specific items are low on stock by running `'list_inventory'`, ensuring timely reordering.

Financial Analyst

Calls `'list_transactions'` repeatedly to audit refund patterns and track revenue sources across different timeframes.

Support Lead

Checks customer profiles or uses `'get_order_details'` to quickly find out why a specific user couldn't access their purchase history.

What Changes When You Connect

-
- 01 Stop jumping between tabs. Instead of opening the admin dashboard to see if a customer paid their balance, just ask your agent about it; it runs `list_transactions` and gives you a clean answer.

 - 02 Never manually compile stock reports again. Running `list_inventory` provides an instant, comprehensive view of your entire product catalog's supply levels.

 - 03 Get full visibility into customer behavior by using `list_member_profiles`. You can analyze demographics or find specific contact points for marketing follow-ups.

 - 04 Simplify shipment tracking. Instead of finding a single order in the queue, you ask your agent to inspect unfulfilled shipments by running `get_order_details` on multiple items at once.

 - 05 Streamline financial audits. By calling `list_transactions`, you get a clean, organized ledger view that lets you track refunds and payments much faster than exporting raw data.
-

Real-World Applications

Investigating Customer Payment Issues

A customer reports they were charged twice. Instead of manually cross-referencing payment sheets, the agent runs ``list_transactions`` and immediately identifies both charges on the correct account.

Marketing Segmentation

You want to run a targeted campaign based on geography. The agent uses ``list_member_profiles`` to pull all customer data and filter it down to only users located in California.

Pre-Sale Inventory Check

You need to know if you can fulfill a large bulk order. You prompt your agent to check ``list_products`` first, then verify available stock with ``list_inventory``, confirming if you have enough units before committing.

Post-Shipment Reconciliation

A client claims an order was never processed correctly. You use the agent to run ``list_orders`` first, then drill down with ``get_order_details``, finding exactly which line items were stuck in the payment queue.

Patterns to Avoid

Trying to find a single customer's entire history

✗ AVOID

Having to navigate from the 'Orders' page, click into the order, then copy-paste the user ID into the 'Members' section just to see their profile.

✓ INSTEAD

Ask your agent to pull data by running ``list_member_profiles`` first. Then, ask it to cross-reference that user against all payments using ``list_transactions``. It handles the linking for you.

Checking stock across multiple product lines

✗ AVOID

Opening six different tabs in the Squarespace backend, clicking into each 'Inventory' section one by one to manually write down current counts.

✓ INSTEAD

Simply ask your agent to run ``list_inventory``. It aggregates all that information instantly and presents it as a clean list.

Verifying payment status for old accounts

✗ AVOID

Searching through archived financial reports or needing an administrator with special permissions just to verify if a historical payment ever cleared.

✓ INSTEAD

Use the ``list_transactions`` tool. It sweeps up all financial movements and tells you exactly when, how much, and if the funds settled.

The Right Fit

You must use this MCP if your job requires compiling data from multiple, disparate pages within Squarespace—for example, listing orders and then checking customer details for each one. Don't use it if you only need to read a single piece of static information (like just the store name). If all you need is a basic list of products without any transactional context, there might be an alternative tool that handles simpler catalog views. However, if your goal involves auditing payments (`list_transactions`), cross-referencing profiles (`list_member_profiles`), or tracking order fulfillment status (`get_order_details`), this MCP gives you the necessary read-only depth.

The endless cycle of logging in and clicking through dashboards.

Today, if you need to know why a specific customer can't access their order details, you start by logging into Squarespace. You click the 'Orders' tab. Then, you search for the customer name. If that fails, you open the 'Members' section, find the profile, and then manually copy-paste IDs to check payment records in another window. It's a tedious dance across five different tabs just to answer one simple question.

With this MCP, you eliminate all that clicking. You ask your agent about the problem—saying, 'Why can't I see order XYZ?'—and it uses its internal access to run those complex checks behind the scenes. It doesn't feel like a database query; it feels like asking an extremely knowledgeable coworker who already has access to everything.

Squarespace MCP: Get orders, inventory, and profiles in one place.

You no longer need to open the 'Orders' section, pull a list of IDs via `list_orders` , and then go check individual details using `get_order_details` . The agent handles that entire workflow internally.

The difference is speed and depth. You get immediate answers synthesized from multiple data sources without having to write a single line of code or navigate a single button click.

It pulls the list, finds the relevant items, and presents the key data points right away.

Squarespace: 6 Data Tools

Use these tools to retrieve specific data points like order details, current inventory levels, or member profiles from your Squarespace e-commerce store.

#	TOOL	DESCRIPTION
01	<code>get_order_details</code>	Fetches all the specifics for one individual order, including line items and shipping info.
02	<code>list_inventory</code>	Returns a list of every product and its current stock count in the store.
03	<code>list_orders</code>	Pulls a comprehensive listing of all commerce orders that have been placed on the site.
04	<code>list_products</code>	Retrieves details for every product available in the store's catalog.
05	<code>list_member_profiles</code>	Generates a list of all customer or member accounts, including contact and demographic data.
06	<code>list_transactions</code>	Provides a chronological record of every financial movement, like payments and refunds.

See It in Action

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

- U** Pull down my customer profiles list. I want to see if any users are from Canada natively mapped resolving via MCP tools.



Pulling data utilizing direct pipeline `list_member_profiles`. Scanned 14 overall valid active objects retrieved and loaded internally.

Filtered Result: I have detected 2 profiles with a shipping/contact locale mapped to Canada ('CA'):

- **Alice Vancouver (alice.v@mail.com)** | ID: P#98s0s1
- **Jean L. (jean.l@domain.ca)** | ID: P#001aX1

I can expand their full line item purchase history tracking backwards if needed.

- U** Use the transaction readout scanner to count how many completely refunded logs appear in the last batch.



Cycling payload natively engaging `list_transactions` auditing backend ledgers internally...

Found **42** recent transactions stored loosely across internal responses. After analyzing inner data vectors, **3 transactions** possess fully executed refund events explicitly attached holding values returned to the buyer payload:

1. **Tx_abc123** (\$40.00 | Jan 02)
2. **Tx_zzz009** (\$15.00 | Jan 04)
3. **Tx_plx444** (\$120.50 | Jan 10)

All remaining logs read normal processed settled positive values.

Frequently Asked Questions

01 Can I use Squarespace MCP to see all my customer addresses?

Yes, you can list member profiles using `list_member_profiles`. This tool gathers the contact and demographic data for all your registered users.

02 How do I check if an order was refunded using Squarespace MCP?

Run `list_transactions` to audit payment history. The agent will filter down the transaction ledger specifically for refund events attached to your orders.

03 What is the best way to list all available products in Squarespace?

Use the `list_products` tool. This gives you a clean, comprehensive listing of every unique product ID and its basic details in your store catalog.

04 Does Squarespace MCP let me see low stock items?

Yes. The agent can combine the `list_products` data with `list_inventory` to filter and report only on products that are near or below your set minimum stock level.

05 Can I pull up a single order's details using Squarespace MCP?

Yes, you call the `get_order_details` tool. This pulls every detail related to one specific transaction, including line items and payment method used.

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 `"squarespace": { "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

Squarespace 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 Squarespace. 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	Squarespace MCP
Server ID	019d760c-9883-7178-bb2f-c6b4671f6dfd
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/squarespace.