MCP SERVER | NO CODE | CLOUD HOSTED

Octoparse MCP

Turn web scraping into conversation.

Octoparse connects your AI agent directly to a full cloud web scraping platform. Run complex extraction jobs, monitor crawler progress in real time, and pull structured data from external websites straight into your chat context. It lets you treat the entire process—from triggering the scrape to analyzing the resulting rows—as one conversational command.

A+ Quality Score 100/100

data-extraction | web-crawling | no-code | automation | data-pipeline | cloud-scraping

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

SCAN TO VIEW

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.

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.

Octoparse MCP

10 tools available
Cloud-hosted on Vinkius

Octoparse turns web crawling into a simple conversation with your AI agent. Instead of dealing with complex API keys or opening multiple browser tabs, you simply tell your agent what data you need from a website. The MCP handles launching the cloud scraping job and keeps track of its progress until it’s done. Once the data is ready, your agent pulls the extracted rows directly into the chat context. You can then ask the AI to summarize competitive pricing or structure an email list based on that newly acquired information. If you're looking for a central place to manage these connections, Vinkius hosts this MCP alongside thousands of other specialized tools, making it easy for your agent to access everything from data extraction to messaging services.

Core Capabilities

01 — Start and stop scraping tasks

You can launch a cloud scrape job when you need fresh data or instantly halt a task that's running too long.

02 — Check live task status

Your agent reports the current progress of any active scraping project, letting you know if it’s running smoothly or stalled.

03 — List all projects and tasks

You can view every folder and individual scraping task configured in your Octoparse account.

04 — Get raw extracted data rows

The MCP fetches the final, structured web rows from a completed job and loads them directly into your agent's working memory for immediate use.

05 — Update scraper parameters

You can dynamically change the core URLs or keywords driving a task without having to rebuild the entire scraping project.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your premium Octoparse API credentials.

02 Next, command your AI agent to perform a specific data action, like starting a task or listing available projects.

03 Finally, the MCP executes the request against Octoparse's cloud servers, delivering the status updates or raw data directly back to your chat window.

The bottom line is that you manage complex web scraping processes using only natural language commands in your preferred AI client.

Built For

This MCP is built for analysts, developers, and marketers who constantly need structured data from the open web. If your job involves collecting competitive intelligence, market research, or building lists of external records, this tool saves hours of manual effort.

Data Engineer

You use it to trigger scheduled pipelines, check extraction states, and dump JSON samples right in your terminal for debugging schemas without switching tools.

Growth Hacker

You quickly run a scraper against platforms like Amazon or LinkedIn, grab the extracted table data, and immediately prompt the AI to format it into an actionable email list.

Business Analyst

You fetch competitive pricing matrices scraped overnight and ask your agent to summarize price drops or identify market trends directly within the conversation window.

What Changes When You Connect

01 Data ingestion is instant. Instead of downloading CSVs, you use the `get_task_data` tool to pull structured rows directly into your agent's context, letting it format or summarize results immediately.

02 Monitoring is transparent. You get real-time status updates using `get_task_status`, so you never waste time wondering if a crawler is stuck or still working.

03 Control is absolute. If a scrape job goes rogue, the `stop_task` tool lets your agent shut it down instantly, saving credentials and compute time.

04 Flexibility matters. Need to change what you are looking for? The `update_task_params` tool lets you shift keywords or URLs driving a task without rebuilding the whole project.

05 Efficiency gains: You can list all tasks with `list_tasks`, giving your agent a complete map of every scraping job, making data retrieval systematic and reliable.

Real-World Applications

Competitive pricing intelligence

A business analyst needs to see price changes across 10 major retail sites. They use the MCP to run multiple scrapers, then feed all the resulting data into the agent via `get_task_data`. The agent then builds a comparative markdown table showing only items that dropped in price by over 20%.

Building lead lists from LinkedIn

A growth hacker wants to build an email list of specific job titles. They use the MCP to start and monitor a targeted scraper, then prompt the agent to pull all collected data using `get_task_data` so the AI can validate the emails against known patterns.

Debugging web schemas

A data engineer needs to verify if a new scrape job captured the correct fields. They use `list_tasks` first, then trigger a specific task run using `start_task`, and finally pull sample JSON via `get_task_data` to debug the schema without leaving their terminal.

Automating market monitoring

A business analyst needs daily pricing reports. Instead of manually re-running tasks, they instruct the agent to check status with `get_task_status`, ensuring the scheduled job ran successfully before requesting the latest data dump.

Patterns to Avoid

Assuming direct API access

✗ AVOID

Trying to manually pass complex URLs or credentials directly into the chat prompt because you think the agent can handle it.

✓ INSTEAD

Always use the structured tools. First, use `get_token` to authenticate, then use `update_task_params` to adjust the target URL before running the task with `start_task`.

Only listing tasks

✗ AVOID

Calling `list_tasks` and thinking that just seeing a list of available projects is enough information for analysis.

✓ INSTEAD

Seeing the list isn't enough. After confirming the task exists, you must use `get_task_data` to actually pull the extracted rows into the agent's context.

Overwriting data prematurely

✗ AVOID

Calling `clear_task_data` when you meant to read the existing results first.

✓ INSTEAD

If you need the data, call `get_task_data` before running any cleanup. Only use `clear_task_data` when you are absolutely certain the old data is useless.

The Right Fit

Use this MCP if your core problem involves extracting structured data from live websites—anything that requires a dedicated web crawler to collect records (e.g., product lists, competitor pricing, directories). This tool handles the entire lifecycle: launch, monitoring, retrieval, and refinement. Don't use it if you need to query an internal database or read a local file; for those needs, look for database connectors or document handling tools. If your goal is

simply messaging or sending alerts based on data *already* acquired, check out communication-focused MCPs instead.

The manual process of competitive intelligence collection is a time sink.

Today, collecting market intel means opening one website, finding the table, right-clicking, and copying it into Excel. Then you repeat that whole cycle for ten competitors, copy-pasting data from dozens of browser tabs, only to spend hours manually cleaning up merged cells and inconsistent formats.

With this MCP, you just tell your agent what kind of data you need. It launches the scraper, tracks its progress until it finishes, and delivers all the structured rows—no messy formatting, no extra clicks—straight into your chat window.

Octoparse: Structured Data Delivery

You skip opening the browser, you skip right-clicking and copying tables, and you skip spending time writing boilerplate API calls to manage task status or credentials.

It's a single command that manages the entire flow. You get perfectly structured data ready for immediate analysis.

Octoparse MCP Tools (10)

These tools let you manage the full lifecycle of web scraping: starting jobs, checking status, updating parameters, and extracting the raw data needed for analysis.

#	TOOL	DESCRIPTION
01	clear_task_data	Deletes all data associated with a specific Octoparse task, useful for cleaning up test runs before starting production crawls.
02	get_task_data	Exports the completed web rows from an Octoparse scraping job so your agent can process them for analysis (limited to 1000 records).
03	get_task_status	Retrieves and reports the current running status of any active task in Octoparse's cloud environment.
04	get_token	Obtains a fresh OAuth 2.0 access token from Octoparse, which is necessary for subsequent API calls.
05	list_task_groups	Lists all top-level folders or groups of tasks within your entire Octoparse account structure.
06	list_tasks	Provides a list of every configured cloud scraping task, including its status and creation date.
07	mark_data_exported	Changes the status of all stored data in an Octoparse task to 'extracted,' confirming it's ready for use.
08	start_task	Initiates a cloud scraping job immediately, changing its status to running within Octoparse.
09	stop_task	Halts any currently running Octoparse cloud task before it completes its cycle.
10	update_task_params	Adjusts the core search URL or specific keywords driving a task, allowing you to scale parameterized bots without reopening the IDE.

See It in Action

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

U Look up task 'LinkedIn Profiles Q4' and tell me how many rows it extracted.

The Cloud Agent confirms the task 'LinkedIn Profiles Q4' finished running successfully and acquired `4523` rows of active data.

U Start my Amazon Price Monitor crawler task now.

Task started. Your 'Amazon Price Monitor' has been queued to the cloud servers and will begin fetching targeted DOM elements shortly.

U Get the data extracted from task 'Real Estate NYC' and format it as a markdown table.

I've fetched the rows successfully. Here is the structured breakdown highlighting the `Address`, `Square Footage`, `Beds`, and estimated `Asking Price`...

Frequently Asked Questions

01 **How do I start scraping with Octoparse MCP?**

You must first obtain an access token using `get_token` and then instruct your agent to use the `start_task` tool, specifying which task group you want active.

02 **What if my scrape fails halfway through Octoparse MCP?**

You can check the current progress using `get_task_status`. If it's stuck, use the `stop_task` tool to halt the job and figure out what went wrong.

03 **Can I change the target website mid-scrape with Octoparse MCP?**

Yes. You don't have to rebuild the whole project; you can use `update_task_params` to dynamically adjust the core search URL or keywords driving the task.

04 **How do I get the data out of Octoparse MCP?**

Use the `get_task_data` tool. This fetches un-exported rows from a completed job, making them available for your agent to analyze and structure immediately.

05 **What is the best way to manage multiple scrapers with Octoparse MCP?**

Use `list_task_groups` and then `list_tasks`. This gives you a full overview of everything configured in your account, letting your agent target specific jobs.

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 `"octoparse": { "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

Octoparse 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 Octoparse. 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	Octoparse MCP
Server ID	`019d75e2-0c97-7183-8c72-172e772531b5`
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/octoparse.