MCP SERVER | NO CODE | CLOUD HOSTED

Import.io Web Data MCP

Turn messy websites into clean, structured data.

Import.io Web Data Extraction MCP lets your AI client scrape and structure data from any website. Run targeted extractors on specific URLs for clean JSON output, initiate massive bulk crawls across multiple pages, or let the Magic API automatically pull tables without pre-configured rules. Monitor job status and download results instantly as CSV or JSON.

A+ Quality Score 100/100

data-extraction | web-crawling | structured-data | json-export | automation | data-pipeline

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/importio-web-data-extraction

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.

Import.io (Web Data Extraction) MCP

10 tools available
Cloud-hosted on Vinkius

Connect this MCP to your agent and take control of web data extraction through natural conversation. You can tell it exactly what you need from any website, whether that's specific product pricing or a list of contacts. Start by triggering predefined extractors on single URLs to get clean JSON right away. Need something bigger? Run large-scale jobs across multiple pages concurrently and track their progress in real time. If the data structure is messy, use the automated Magic API to pull tables without needing setup. Once the work's done, you can retrieve results as structured JSON or CSV files, ready for your spreadsheet program. Everything runs through Vinkius, giving your agent access to thousands of other tools alongside this one.

Core Capabilities

01 — Run Targeted Data Extracts

Trigger specific, predefined data extractors on single web pages to pull clean JSON content directly into your workflow.

02 — Execute Bulk Crawls

Start large-scale scraping jobs across many pages at once and monitor the progress of the entire crawl job.

03 — Extract Unstructured Data

Use the automated Magic API to identify and pull tabular data from any website, even if you haven't set up a specific extractor for it.

04 — Track Job Statuses

Poll ongoing extraction runs or bulk crawl jobs to check their current state, success rates, and total pages processed.

05 — Export Structured Files

Retrieve final extraction results in either structured JSON format or ready-to-use CSV text for immediate processing.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/importio-web-data-extraction** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Import.io API Key.

02 Tell your agent what you want: whether it's running a targeted extractor, starting a massive crawl, or using the Magic API.

03 Your agent will manage the job, track its progress (like pages processed), and then retrieve the final data in JSON or CSV format.

The bottom line is you can turn unstructured website content into clean, usable data formats without writing any scraping code.

Built For

This MCP is for analysts and researchers who spend too much time copying raw web data into spreadsheets. If your job involves monitoring competitors or collecting market intelligence from multiple sites, this connector saves hours of manual effort.

Market Researcher

Runs large-scale competitor audits and monitors pricing changes across dozens of websites without needing to write custom scraping scripts.

Data Analyst

Automates the collection of market data, building structured datasets from diverse online sources using natural language prompts.

Product Manager

Verifies and monitors data schemas by running extraction tests across multiple project sites to ensure data integrity before launch.

What Changes When You Connect

01 Automate market intelligence collection by running predefined extractors via `run_extractor` against specific competitor product pages. You get the exact data points you need in JSON format every time.

02 Handle massive web audits without writing a single line of code. Use `start_crawl` to monitor progress across hundreds of URLs, letting your agent know when the full dataset is ready.

03 Bypass setup entirely with the Magic API (`run_magic_api`). If you just need pricing tables from a random site and don't have an extractor built, this feature gets it for you instantly.

04 Keep track of everything. You can check job status using `get_extractor_status` or monitor your budget by running `account_usage` , so you never hit a credit wall when you need data most.

05 Get the output in exactly what you need: use `download_csv` to instantly export results for spreadsheet processing, skipping the manual copy-pasting steps entirely.

Real-World Applications

Monitoring Competitor Price Changes

A market researcher needs daily pricing data from a rival's product catalog. Instead of manually entering URLs into a scraper, they ask their agent to use the `run_extractor` tool with the specific 'Product Pricing' extractor against all 50 competitor SKUs. The results are compiled and delivered as structured JSON.

Building an Industry Directory

A business developer needs contact details (email, phone) from a list of websites that don't have standardized data. They ask their agent to use the Magic API (`run_magic_api`) across all 10 sites and then compile the resulting data into one clean CSV file for follow-up.

Auditing Website Content Scale

A content strategist wants to see how many articles a competitor has published over two years. They ask their agent to use `start_crawl` across the main blog section, monitor the progress using `get_crawl_status`, and get a final count of pages processed.

Validating Data Schema for New Products

A product manager needs to confirm that all new product listing sites follow the same data format. They run several targeted extracts, check the output using `get_extractor_data`, and ensure the JSON keys are consistent across all sources.

Patterns to Avoid

Trying to scrape dynamic JavaScript content

✗ AVOID

Asking your agent to 'Just get me the data from this site' without specifying that you need structured tables, leading to vague or incomplete results.

✓ INSTEAD

Don't rely on general prompts. Always use a targeted tool like `run_extractor` for known structures, or specifically invoke `run_magic_api` when the site content is unpredictable.

Assuming all data comes as clean JSON

✗ AVOID

Getting raw text dumps from an extract run and then having to manually parse out tables and specific fields.

✓ INSTEAD

If you anticipate needing spreadsheet input, use the `download_csv` tool. It formats the output correctly for immediate use in Excel or Google Sheets.

Ignoring API usage limits

✗ AVOID

Running multiple large jobs without checking your available credits and suddenly getting an 'Insufficient Funds' error mid-process.

✓ INSTEAD

Always start by running `account_usage` to know exactly how many API credits you have left for the month. This prevents costly interruptions.

The Right Fit

Use this MCP if your core need is turning unstructured, messy web content into clean, structured data formats like JSON or CSV. It's ideal for market research and large-scale audits because it handles both targeted extraction (predefined rules) and opportunistic extraction (Magic API). However, don't use this if you need to interact with a website's backend—it only reads public content. If your goal is complex data manipulation *after* the scrape, you'll need a dedicated database connector; this MCP handles the gathering

phase only. Never use it if you just want basic text scraping; stick to `run_extractor` for structured results.

The headache of copying web tables into spreadsheets

You know the routine: finding a competitor's price list or a product spec sheet online. You click, you right-click, and you copy the data. Then, when you paste it into your spreadsheet, half the columns are mashed together, other formatting breaks, and you spend twenty minutes just separating the SKU from the name.

With this MCP, you simply tell your agent which website needs monitoring and what data points matter. The system uses tools like `run_extractor` or `run_magic_api` to read the page's underlying structure, pulling out perfectly clean, separated data that lands directly into a usable JSON or CSV format.

Get structured web data with Import.io Web Data MCP

Manual scraping scripts require you to write code for every single site layout change, and monitoring requires juggling multiple tabs just to see if the job succeeded or failed.

Now, your agent manages the whole process. You initiate a crawl with `start_crawl`, check its status using `get_crawl_status` when needed, and retrieve all results later via `get_crawl_data`. It's automated monitoring without the code overhead.

Import.io (Web Data Extraction) with 10 Tools

These tools let you manage the full lifecycle of web scraping, from starting a bulk crawl to downloading final structured data outputs.

#	TOOL	DESCRIPTION
01	`get_crawl_data`	Pulls the final, organized JSON output after a large crawl job has finished.
02	`get_crawl_status`	Checks if an ongoing bulk crawl is still running and how many pages it's processed so far.
03	`download_csv`	Downloads the extracted data as plain CSV text, ready to paste into a spreadsheet.
04	`get_extractor_data`	Retrieves structured JSON data from a single extraction run once that job has completed.
05	`list_extractors`	Lists all the custom extractors already set up in your Import.io account so you know which ones to use.
06	`run_magic_api`	Runs an automated scan against a URL, automatically pulling out tables and structured information without any setup.
07	`run_extractor`	Starts a specific, predefined data extraction job on a single website URL.
08	`start_crawl`	Initiates a large-scale bulk crawling operation across multiple pages at the same time.
09	`get_extractor_status`	Checks the current state of any single extraction job, showing if it's running, done, or failed.
10	`account_usage`	Reports how many API credits you've used this month against your subscription limit.

See It in Action

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

U Run extractor 'ext-123' against 'https://example.com/products'

Extraction run started. Run ID: 'run-98765'. I'll monitor the status for you. It usually takes a few seconds to process dynamic pages. Would you like me to fetch the results once it's completed?

U List all extractors in my Import.io account

I've found 5 extractors in your account: 'Amazon Scraper' (ext-001), 'Competitor Pricing' (ext-002), 'Real Estate Leads' (ext-003), 'Stock Tracker' (ext-004), and 'News Monitor' (ext-005). Which one would you like to run?

U Check my monthly API credit usage

You've used 12,450 credits out of your 50,000 monthly limit (24.9%). You have 37,550 credits remaining for this billing cycle. Your account health is excellent.

Frequently Asked Questions

01 **How does Import.io Web Data MCP handle websites that change often?**

The system uses predefined extractors for stable data points, but if a site layout changes completely, you can use the Magic API to try and pull general structured tables automatically.

02 **Can I run Import.io Web Data MCP on private sites?**

No. This MCP is designed for public web data extraction using standard scraping methods. It cannot access protected or login-required content.

03 **What's the difference between `run_extractor` and `start_crawl`?**

`run_extractor` targets a single, specific page with known data points. `start_crawl` is for bulk jobs across multiple pages or sections of a large website.

04 **What information does the account_usage tool provide?**

It tells you exactly how many API credits you've consumed this billing cycle and what your remaining credit balance is, helping manage your budget.

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 `"importio-web-data-extraction": { "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

Import.io (Web Data Extraction) 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 Import.io (Web Data Extraction). 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	Import.io (Web Data Extraction) MCP
Server ID	`019d75b8-ca26-7143-b769-86fe16a459b0`
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/importio-web-data-extraction.