MCP SERVER | NO CODE | CLOUD HOSTED

Browserless MCP for AI Agents

Extracting and capturing structured web assets from complex sites

Browserless lets your AI agent automate web tasks by controlling headless Chrome directly. You can take high-quality screenshots, generate polished PDFs from any URL, extract fully rendered content from JavaScript sites, and run targeted scraping requests—all through natural conversation.

C Quality Score 78.57/100

headless-chrome | pdf-generation | screenshot-capture | web-scraping | automation | cloud-browser

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

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.

Browserless MCP

8 tools available
Cloud-hosted on Vinkius

When you connect this MCP to your preferred AI client, it gives your agent the ability to interact with the live web browser. You stop treating websites as static text blocks; instead, you treat them like actionable data sources. Need a visual record? Just ask for a screenshot of a specific page or an element. Want reports? The system converts entire webpages into clean, professional PDFs. And if you need structured information, it can run custom scraping scripts to pull out exactly what you're looking for—like all product titles or key dates. Since Vinkius hosts this MCP in the catalog, your agent gets immediate access to these web automation tools alongside everything else you use.

Core Capabilities

01 — Capture Web Visuals

Take detailed screenshots of any URL, including options for full-page captures.

02 — Generate Documents from URLs

Convert the content of an entire webpage into a polished PDF document.

03 — Extract Rendered Web Content

Retrieve the complete HTML code, even from complex websites that rely heavily on JavaScript to load their data.

04 — Targeted Data Scraping

Run specific scraping requests by defining element selectors and extracting only the necessary pieces of data.

05 — Monitor Browser Activity

List currently active browser sessions or check general service health status.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your personal Browserless.io API Token.

02 Your AI client connects to the Vinkius catalog, recognizing this web automation capability.

03 You issue a simple command through natural language (e.g., 'Take a screenshot of X URL'), and the agent executes the necessary headless Chrome tasks.

The bottom line is that your AI client runs complex browser commands—like taking screenshots or scraping data—without you having to open a single tab yourself.

Built For

Web Developers who need quick debugging visuals, QA Testers running visual regression tests, and Content Managers needing structured web reports. If your job involves looking at websites and extracting information from them, you need this MCP.

QA Tester

Automating visual regression captures or listing active sessions to verify if a webpage is displaying correctly across different states.

Web Developer

Debugging rendered content by retrieving the full HTML of complex sites, or taking screenshots for documentation without manual browser work.

Content Manager

Generating polished PDF versions of online articles or reports so they can be shared easily outside of a web link.

What Changes When You Connect

01 Generate polished reports instantly. Instead of manually printing or saving a webpage, use the `generate_pdf` tool to create professional PDF documents directly from your agent.

02 Bypass JavaScript roadblocks. The `get_page_content` tool retrieves the full HTML source code, letting you access data from modern, dynamic websites that usually fail with basic scrapers.

03 Automate visual verification. Use `take_screenshot` to grab high-quality images of specific URLs or entire pages for documentation and QA checks without touching a physical browser.

04 Isolate specific data points. If you only need the names and prices from an e-commerce page, the `run_scrape` tool lets you target those exact elements using selectors.

05 Maintain oversight. The `list_active_sessions` tool lets your agent check if background automation tasks are still running or have failed.

Real-World Applications

Documenting a Client Website

A Content Manager needs to provide a client with a report on their latest marketing blog post. They ask the agent to run `generate_pdf` on the URL, getting a single, clean, branded document ready for immediate review.

Debugging Front-End Code

A Web Developer suspects a JavaScript component isn't loading data correctly. They ask the agent to run `get_page_content` on the page, allowing them to examine the full rendered HTML and find the missing element.

Gathering Competitive Data

A market analyst needs a list of product titles from 5 competitor shop pages. They instruct the agent to use `run_scrape`, specifying selectors for 'product title' across all five URLs, compiling one structured list.

QA Visual Regression Testing

A QA Tester runs a workflow that uses `take_screenshot` on key landing pages before and after a code deployment. They can compare the resulting images to ensure no visual elements shifted unexpectedly.

Patterns to Avoid

Relying only on basic scraping

✗ AVOID

Attempting to scrape data from a modern news site using simple text extraction, which fails because the content is loaded by JavaScript after the initial page load.

✓ INSTEAD

Use `get_page_content` first. This retrieves the fully rendered HTML source code that includes all JavaScript-loaded elements, guaranteeing you capture the complete data set before scraping.

Manually capturing multiple pages

✗ AVOID

A developer having to open five tabs and take a full-page screenshot for documentation purposes.

✓ INSTEAD

Use the `take_screenshot` tool. You can command your agent to capture screenshots of multiple URLs sequentially, automating what used to be a tedious click-and-save process.

Missing usage context

✗ AVOID

Running scraping jobs without knowing if the service is overloaded or if the API token has expired.

✓ INSTEAD

Always check `check_system_health` and review your limits using `get_usage_stats` before starting a major automation run to prevent failures.

The Right Fit

Use this MCP when your primary goal is interacting with the *presentation layer* of a website. You need to capture what the user actually sees—be it an image, a PDF report, or structured data derived from complex rendering. Don't use it if you only need to interact with pure APIs; those are better handled by direct API calls. However, use this MCP for anything that requires simulating a real browser visit. Specifically, if your task involves generating PDFs via `generate_pdf` or needing the full rendered content using `get_page_content`, this is the right tool. If you only need to list users or manage internal records, then another, non-web focused MCP will be more appropriate.

Browserless MCP for AI Agents: Overcoming Web Content Rendering Challenges

Today, pulling data from a modern website means navigating several roadblocks. You open the page, but the actual article text—the thing you need—is loaded by JavaScript only after the initial page load. Your agent might pull the basic HTML structure, leaving out key paragraphs, images, or interactive elements entirely. This forces manual checks and copy-pasting across multiple sources.

With this MCP, your agent treats the web like a live browser session. You simply ask it to get the full content of an article, and it uses `get_page_content` to pull every piece of rendered data, giving you the complete, accurate source material you need for analysis.

Browserless MCP for AI Agents: Automating Web Data Extraction and Visualization

Manual web reporting involves endless clicking. You might have to navigate five pages, copy the title from one, download a PDF from another, and take screenshots of the final summary page—all before you can share the report.

Now, your agent orchestrates this entire workflow. It handles taking high-quality visual captures with `take_screenshot`, compiles the whole thing into a single document using `generate_pdf`, and delivers a complete package ready to send. The manual clicks are gone.

Browserless: 8 Tools for Web Content Extraction and Automation

These tools let your agent perform specific tasks like taking screenshots, generating documents, scraping data, or checking service health directly from the web browser environment.

#	TOOL	DESCRIPTION
01	`check_system_health`	Checks if the Browserless service is currently running and available for use.
02	`generate_pdf`	Creates a downloadable PDF document using all the content from a specified URL.
03	`get_account_config`	Retrieves your specific account settings and configuration details for verification.
04	`get_page_content`	Fetches the fully rendered HTML source code from a given URL, including JavaScript-loaded elements.
05	`get_usage_stats`	Retrieves metrics about your account usage history and current quotas.
06	`list_active_sessions`	Lists all browser sessions that are currently running through the service.
07	`run_scrape`	Executes a custom scraping script based on element selectors to extract specific data points.
08	`take_screenshot`	Captures an image of any URL using headless Chrome, supporting full-page views.

See It in Action

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

U Can you grab a full-page screenshot of that article at example.com/blog/post-1?

Screenshot Status: ✅ Complete

URL Captured: `https://example.com/blog/post-1`
Capture Type: Full Page
Dimensions: 1920 × 3540 pixels

[Image Placeholder: High-res screenshot of the article]

The image is attached, showing the header through to the comments section. Let me know if you need a cropped version.

U I need product titles and prices from this shop page.

I ran the scraping script on that URL and pulled out the data based on your request. Here are the top 5 results:

PRODUCT TITLE	PRICE
Leather Jacket, Deluxe	$499.00
Denim Jeans, Slim Fit	$89.99
Cotton T-shirt, Basic	$19.99
Running Shoes, Ultra X	$125.00
Wool Scarf, Winter Blend	$55.00

Would you like me to export this list into a CSV file?

U Turn the entire content of this research paper into a PDF.

PDF Generation Status: ⚙ Processing...

Please wait while I compile all assets and formatting from `https://example.com/research-paper`.

Success!
Your document is ready for download: [Download Link].
The resulting file, 'Research Paper.pdf', maintains the original formatting including tables and embedded images.

Frequently Asked Questions

01 **How does Browserless MCP handle websites that load content with JavaScript?**

It retrieves the fully rendered HTML source code, meaning it doesn't just grab what appears initially. It processes the site as if a user had opened it in Chrome, allowing you to access dynamic data and interactive elements.

02 **Can I use Browserless MCP to make PDF reports from URLs?**

Yes, you can use the generate_pdf tool. Simply provide the URL, and the agent returns a polished, downloadable PDF document that preserves the original page's formatting.

03 **Is Browserless MCP only for taking screenshots?**

No. It's much more than just visual captures. You can use it to scrape structured data, get full HTML content, and create documents all through natural language conversation.

04 **What if I need a list of specific items from a webpage?**

You can run custom scraping scripts. By telling the agent exactly what element selectors to look for (like product names or dates), it pulls out only that information and gives you a clean list.

05 **Are there limits when using Browserless MCP?**

You can check your usage limits by running the get_usage_stats tool. This provides clear metrics on your current account status so you know how much capacity you have for automation runs.

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 `"browserless": { "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

Browserless 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 Browserless. 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	Browserless MCP
Server ID	`019d7564-bba1-71b1-b3a3-d36b51cd5852`
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/browserless.