MCP SERVER | NO CODE | CLOUD HOSTED

Browserless (Playwright Cloud) MCP for AI Agents

Scraping complex Single Page Applications and extracting dynamic web data

Browserless (Playwright Cloud) connects your AI client to a remote, scalable browser cluster, allowing it to scrape and interact with complex websites. This MCP handles Single Page Applications (SPAs), bypassing anti-bot measures like Cloudflare or Data Dome. Instead of just fetching raw text, your agent runs full JavaScript lifecycles, letting you extract fully rendered HTML, high-fidelity PDFs, or specific elements even when the content loads dynamically.

A+ Quality Score 100/100

headless-browser | test-automation | web-scraping | chromium | browser-testing | cloud-execution

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

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 (Playwright Cloud) MCP

10 tools available
Cloud-hosted on Vinkius

Modern websites are tricky. They don't hand over clean data through simple API calls; they load it using complex JavaScript and React frameworks, often protected by anti-bot measures. This MCP gives your AI agent a full browser sandbox—a legitimate remote Chromium instance—to tackle those sites head-on. You no longer have to worry about whether the content is loaded correctly or if basic fetching methods are blocked. Your agent can execute custom JS functions, simulating clicks on modals before extracting data, or waiting until specific selectors appear. It's like giving your AI a full set of hands and eyes for any browser page. When you connect this MCP via Vinkius, you gain the ability to treat every website as if it were running locally in a dedicated testing environment, guaranteeing accurate data capture whether you need raw HTML, a PDF printout, or just a screenshot.

Core Capabilities

01 — Render Full JavaScript DOM

The agent executes full JavaScript lifecycles on the target page, ensuring that complex Single Page Applications (SPAs) fully load their content before extraction.

02 — Extract Specific Visual Elements

You can instruct your AI to pinpoint and pull out data using explicit CSS selectors rather than scraping large chunks of text.

03 — Capture Page Screenshots and PDFs

The MCP spawns a temporary browser instance to capture high-resolution full-page screenshots or generate print-perfect PDF documents automatically.

04 — Bypass Anti-Bot Defenses

Requests run through stealth plugins and can be routed via proxies, helping your agent avoid detection by major website security systems like Cloudflare.

05 — Execute Custom Code Blocks

The MCP exposes the raw Chrome DevTools Protocol, letting you inject custom JavaScript code to manipulate or test the page state before extraction.

One Click on Vinkius — From Prompt to Execution

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

01 First, append the Browserless integration details to your AI client settings and provide your secured HTTP Token.

02 Next, prompt your agent with a request that requires full web rendering—for example, asking it to capture a screenshot or extract data from a dynamic page.

03 Your client sends the command to the remote browser cluster, which executes the JavaScript lifecycle, captures the requested artifact (e.g., HTML, PNG), and returns the resulting clean data to your agent.

The bottom line is that you get reliable access to a fully rendered web page environment directly within your AI workflow, regardless of how complex or protected the target site is.

Built For

Anyone who needs data from modern websites—especially those built with React, Vue, or Angular—needs this. If you're tired of scraping scripts that break because a button click requires a JavaScript action first thing in the morning, this is for you.

Data Engineer

You use this to extract data from complex frontend dashboards or e-commerce sites where standard API calls don't exist. You can reliably pull content using `get_html_content`.

QA Automation Tester

You instruct the AI to simulate user journeys, running a series of actions like clicking modals or waiting for selectors before capturing an image with `get_screenshot` to test builds.

Security Researcher

You need to test URLs that might trigger bot detection or require specific proxy endpoints. You can route suspicious queries through isolated cloud environments using `scrape_with_proxy`.

What Changes When You Connect

01 Guaranteed full rendering: Use `get_html_content` to pull the entire DOM structure, even from sites that rely heavily on JavaScript hydration.

02 Visual confirmation: Need proof? Run `get_screenshot` for a precise image capture of the page state at any moment.

03 Advanced data targeting: Instead of dumping everything, use `scrape_elements` to grab only the specific piece of information you need from a large table or block of text.

04 Circumvent security blocks: Use `scrape_with_stealth` or `scrape_with_proxy` when hitting sites that actively challenge automated access. This gets data where simple scrapers fail.

05 Action-based scraping: If the data is hidden behind a modal, use `scrape_with_js` to programmatically click it open before extraction.

Real-World Applications

Collecting Quarterly Pricing Data from E-commerce Sites

A market analyst needs the price list for 50 different products, but each product page loads prices dynamically. They prompt their agent to use `get_html_content` and then `scrape_elements` repeatedly across a catalog index to build a complete spreadsheet.

Testing New Web Features Before Launch

A QA team member tells the AI to run through a simulated user journey, forcing it to execute clicks and waits. The agent uses `get_screenshot` multiple times with `scrape_with_wait` to confirm that every step in the workflow renders correctly.

Archiving Legal Documents from Government Sites

A compliance officer needs a permanent record of an online policy document. They instruct the agent to visit the URL and use `get_pdf` to generate a universally readable, print-perfect PDF copy.

Retrieving Data from Private or Protected Dashboards

A data scientist needs metrics locked behind a login wall. They ask the agent to run through proxy routes using `scrape_with_proxy` and execute custom scripts with `run_custom_function` to pull raw, difficult-to-access numbers.

Patterns to Avoid

Using simple API fetching for SPAs

✗ AVOID

Trying to get the full content of a modern blog page using only basic network requests. The result is usually an empty or incomplete shell because the actual data loads seconds later via JavaScript.

✓ INSTEAD

You need to run the entire browser through its lifecycle. Use `get_html_content` so your agent can wait for and capture the fully rendered DOM structure, ensuring all content appears.

Ignoring bot detection measures

✗ AVOID

Running a scraper that immediately fails when it hits Cloudflare or any WAF challenge. The script stops before getting to the actual data.

✓ INSTEAD

Always use `scrape_with_stealth` and consider wrapping requests with `scrape_with_proxy`. This makes your agent look like a human user, bypassing common bot defenses.

Extracting data before rendering finishes

✗ AVOID

Asking the AI to grab text immediately after landing on a page that requires clicking a 'Load More' button first. The scraper only sees the initial content and misses everything else.

✓ INSTEAD

Tell your agent to use `scrape_with_js` or `scrape_with_wait`. This lets you simulate user actions, like clicking, before attempting extraction.

The Right Fit

Use this MCP if the data source is a modern Single Page Application (SPA) that requires JavaScript execution to load content. If your problem involves capturing screenshots of specific UI states or generating print-ready PDFs from dynamic web pages, this is your tool. Don't use it if you are dealing with simple, static HTML or standard REST API endpoints; for those cases, a basic JSON fetcher will be faster and simpler. However, even when using basic fetching methods, running the data through `get_html_content` ensures maximum compatibility and robust parsing of complex structures.

Browserless (Playwright Cloud) MCP: Solving Complex SPA Data Extraction

Today, gathering web data is a manual nightmare. You open the site, click through tabs to find all the necessary information, and then you have to copy-paste everything into a spreadsheet or Notion page. If the website uses any kind of modern framework—like React—the content often only appears after a delay or requires specific interaction.

With this MCP, your agent handles that entire tedious process. You simply ask for the data, and it runs in a remote browser environment to mimic human behavior. The result is clean, comprehensive HTML that's ready to be structured immediately within your workflow.

Browserless (Playwright Cloud) MCP: Advanced Web Scraping Workflow Automation

You used to have to write complex scraping scripts just to handle basic anti-bot measures, constantly debugging why your script failed because a site updated its security headers. This meant more time writing code and less time analyzing the data.

Now, you can offload that complexity. You tell your agent what you need, and it handles the browser setup, stealth protocols, and element waiting in the background. It's reliable web automation without needing a deep expertise in headless browser infrastructure.

Browserless (Playwright Cloud): 10 Web Scraping & Automation Tools

Use these tools to get raw HTML, high-resolution screenshots, PDFs, or run custom JavaScript on any dynamic website.

#	TOOL	DESCRIPTION
01	`get_html_content`	Retrieves the fully rendered HTML structure of a web page, even if it relies on JavaScript to load content.
02	`send_custom_payload`	Allows you to send arbitrary JSON data directly to any Browserless REST endpoint for advanced interactions.
03	`run_custom_function`	Executes specific, custom JavaScript functions within the isolated environment of the remote browser instance.
04	`scrape_with_js`	Injects and runs a script before scraping, letting you simulate actions like clicking buttons to reveal hidden data.
05	`get_pdf`	Generates a high-quality PDF document from the currently loaded webpage content.
06	`scrape_with_proxy`	Routes all scraping requests through a specified proxy server to mask your true origin IP address.
07	`scrape_elements`	Extracts only the data contained within specific, user-defined CSS selectors from the page.
08	`get_screenshot`	Takes a complete visual snapshot of the webpage, useful for auditing or documenting UI states.
09	`scrape_with_stealth`	Applies advanced plugins to mimic human browsing patterns, bypassing common bot detection systems.
10	`scrape_with_wait`	Pauses the scraping process until a specific element appears on the page, preventing premature data extraction.

See It in Action

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

U Can you take a screenshot of this product listing page? I need to audit how it looks on mobile.

Screenshot Audit: Vinkius Landing Page

- **Device:** iPhone 14 Pro (Portrait)
- **Dimensions:** 390px x 844px
- **Status:** Rendered successfully. All elements visible.

[Image Attached - Full-Page Screenshot]

The screenshot confirms the mobile layout is clean and responsive. No content clipping detected.

U I need the full text of the article, but it only appears after I click the 'read more' button. Can you extract that content?

Article Content Extraction Complete

The original short summary was successfully expanded by simulating a user click on the 'Read More' anchor tag. The extracted text is comprehensive and stable.

- **Key Takeaway:** [First paragraph of full article]
- **Section 2:** [Detailed points from the middle section]
- **Conclusion:** [Final summary statement]

The data looks clean, ready for summarization.

U Can you scrape the name and price of all items listed on this page using a proxy from Germany?

Scraped Item List (German Proxy)

PRODUCT NAME	PRICE	STATUS
Widget Alpha	€19.99	Active
Gizmo Beta	€45.00	Low Stock
Thing Gamma	€7.50	Pre-order

Extraction was performed successfully using a residential German endpoint, bypassing rate limits and ensuring clean data capture.

Frequently Asked Questions

01 **What is the difference between using Browserless (Playwright Cloud) and a simple API call?**

A simple API call only retrieves static data; it can't run JavaScript. This MCP runs a full browser, meaning you get access to content that loads dynamically—like personalized dashboard metrics or content hidden behind click actions.

02 **Can I use Browserless (Playwright Cloud) to scrape sites protected by anti-bot measures?**

Yes. The MCP includes tools like `scrape_with_stealth` and proxy support, which helps your agent emulate human behavior and bypass common detection systems from major security services.

03 **How do I get a reliable PDF document from a complex website?**

Simply ask the MCP to generate a PDF. It simulates printing the page perfectly, capturing all visible elements and ensuring the final document looks exactly like it would if you printed it directly in a browser.

04 **Is Browserless (Playwright Cloud) better for QA testing than traditional automation tools?**

For AI-driven workflows, yes. It allows your agent to run full user journeys—like clicking through a multi-step form and capturing screenshots of each step—making it easier to audit complex web flows.

05 **Does Browserless (Playwright Cloud) handle Single Page Applications (SPAs)?**

Absolutely. SPAs are built on JavaScript, which this MCP fully supports. It ensures that even if the content loads slowly or in chunks, your agent captures the final, complete version of the page.

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-playwright-cloud": { "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 (Playwright Cloud) 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 (Playwright Cloud). 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 (Playwright Cloud) MCP
Server ID	`019d75f7-67ce-7065-9b26-ce3b0b600438`
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-playwright-cloud.