MCP SERVER | NO CODE | CLOUD HOSTED

ScreenshotOne MCP

Automate visual content auditing and archiving.

ScreenshotOne MCP helps your AI agent capture, audit, and archive websites visually. Instead of opening a browser, your agent can instantly grab full-page screenshots or specific elements on any URL. It also generates professional PDFs from web pages, letting you check site designs, monitor content changes, or gather metadata without ever touching a manual workflow.

A+ Quality Score 100/100

screenshot-api | pdf-generation | web-automation | visual-auditing | image-processing

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

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.

ScreenshotOne MCP

6 tools available
Cloud-hosted on Vinkius

Managing visual content across multiple websites is usually a total pain. You have to open tabs, resize windows, and copy-paste screenshots just to verify if a button moved or if the layout changed. This MCP connects your agent directly to high-resolution web capture tools. When you use it through Vinkius, your agent handles the whole process; you simply ask for what you need in plain language.

Need to audit how a site looks on mobile vs. desktop? Your agent can take screenshots of specific viewports or isolated elements. Want to archive an article? It grabs the full page and compiles it into a shareable PDF. Even if you just want to know the title and size of a page, your agent gets that data immediately without needing a full capture. This MCP turns complex visual auditing tasks—the kind of thing that used to take hours of manual clicking—into simple conversations with your AI client.

Core Capabilities

01 — Capture full-page site screenshots

The agent takes an image showing the entire website, from top edge to bottom edge.

02 — Audit specific page elements

You can capture just a small section of a webpage, like a contact form or a specific banner, without grabbing everything else.

03 — Create PDF reports from URLs

The agent generates clean, professional PDFs that contain the full content of any given website address.

04 — Get site metadata instantly

You can pull core data like the page title or overall file size without generating a single image or PDF.

05 — Capture specific viewports

The agent takes screenshots scaled to precise dimensions, simulating different device screens.

One Click on Vinkius — From Prompt to Execution

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

01 Connect your AI client to the ScreenshotOne MCP using your access key.

02 Tell your agent exactly what you need—for example, 'Give me a PDF of this URL' or 'Take a full-page screenshot of X'.

03 Your agent uses the tools to perform the capture and returns the high-quality image link or generated file.

The bottom line is that you talk to your AI client, and it does the visual work for you.

Built For

This MCP is essential for anyone who needs to visually verify digital content or audit websites without manually opening a browser. It's perfect for Quality Assurance engineers checking layout changes, designers verifying element alignment, and archivists rapidly compiling visual reports.

QA Engineer

You use it to check website layouts across different versions or screen sizes, making sure that a recent deployment didn't break any key elements.

UX/UI Designer

You audit specific pages against design specs. You might need to capture a component screenshot to confirm the padding or spacing is correct before signing off on a build.

Content Archivist

You generate PDF copies of important online articles or reports so that your team has an offline, immutable record of the content at a specific time.

What Changes When You Connect

01 Verify design consistency instantly. Instead of manually resizing your browser window to check mobile vs. tablet views, use the `take_viewport_screenshot` tool to capture precise images for comparison.

02 Simplify content archiving. Generate professional PDFs from any URL using the `generate_pdf` tool, giving you a reliable, shareable record without needing manual screen grabs.

03 Focus on what matters with targeted shots. When auditing specific components, use `take_element_screenshot` to isolate and capture only the necessary CSS element.

04 Get data fast, no pictures needed. Use `get_page_metadata` to pull titles, descriptions, and page sizes in a single API call, skipping the visual step entirely.

05 Capture the full context every time. The `take_full_page_screenshot` tool ensures you get a complete view of long-form content or complex site layouts, eliminating partial captures.

Real-World Applications

The QA team needs to check for visual regressions.

A developer pushes an update and the QA engineer asks their agent: 'Take a full-page screenshot of the checkout flow, then take three specific element screenshots (the payment button, the error message box) to verify the layout.' The agent uses `take_full_page_screenshot` followed by multiple calls to `take_element_screenshot`, giving instant visual proof.

The marketing team needs to archive a press release.

Instead of printing or taking messy screenshots, the archivist asks their agent: 'Generate a PDF from this URL.' The agent uses `generate_pdf` and provides a clean, professional document ready for immediate storage in a knowledge base.

The developer needs to check site performance quickly.

Before writing any code, the developer asks their agent: 'What is the page size of the homepage?' The agent uses `get_page_metadata`, returning the file metrics immediately so development can proceed without waiting for a full visual capture.

The designer needs to see how content looks on different devices.

The designer asks their agent: 'Show me what this page looks like in both 375px and 1440px viewports.' The agent uses `take_viewport_screenshot` multiple times, giving the designer precise visual data for design sign-off.

Patterns to Avoid

Trying to grab everything with one shot

✗ AVOID

Asking your agent to 'Capture this whole website and save it.' This is too vague. You might get a partial image, or the agent may fail because you didn't specify if you want a PDF, an element screenshot, or the entire page.

✓ INSTEAD

Be specific: Use `take_full_page_screenshot` when you mean everything, or use `generate_pdf` if you need an archival document. If you only care about one section, call `take_element_screenshot` with the CSS selector.

Assuming metadata is enough

✗ AVOID

The user asks: 'What's on this page?' and expects a full visual answer. Simply using `get_page_metadata` only returns titles and sizes, which won't satisfy the request.

✓ INSTEAD

To know what's visually there, you need to use one of the capture tools. Start with `take_screenshot` for a general view, then follow up if you need more detail.

Over-relying on basic screenshots

✗ AVOID

Using the simple `take_screenshot` tool when you actually need to compare how content looks on a phone versus a desktop.

✓ INSTEAD

For device comparisons, always use `take_viewport_screenshot`. It forces the capture to adhere to specific dimensions, giving accurate representation of different screen sizes.

The Right Fit

Use this MCP if your primary need is visual validation or content archiving. Specifically, you should use it when: 1) You need a complete record (use `take_full_page_screenshot` and then `generate_pdf`). 2) You are performing QA/Design checks across

different devices (use `take_viewport_screenshot` or `take_element_screenshot`). 3) You need to capture content without the overhead of opening any browser (use `get_page_metadata`).

Do NOT use this MCP if: 1) Your goal is simply to check a file's existence. Use a basic data retrieval tool instead. 2) You are trying to scrape massive amounts of unstructured text that needs deep NLP processing; you need specialized scraping tools for that, not just visual captures. If your task involves structured data extraction (like pulling names and dates), look into dedicated form-parsing MCPs.

Manual content auditing is a time sink.

Think about how you spend an hour checking if a new website layout matches the old one. You open the site, resize the browser to check mobile views, take a screenshot, upload it somewhere, repeat for desktop, then maybe zoom in on a specific widget and take another picture. It's a cycle of tabs, clicks, copy-pasting, and endless manual comparison.

With this MCP, you just tell your agent: 'Show me the full page capture at 375px width, and also give me an element screenshot for the footer.' You get all those high-quality image links back instantly. The whole process shrinks from half an hour of clicking into a single command.

ScreenshotOne MCP: Visual data in conversation.

The manual steps that vanish include the need to open browser dev tools, manually adjust viewports, and then use an external tool just to grab a clean PDF version of the content you captured. That's all gone.

You interact with your agent via natural language; it handles the complexity of multiple visual capture types—from `take_element_screenshot` to `generate_pdf`. You stop managing screenshots and start getting answers.

ScreenshotOne: 6 Visual Data Tools

These tools give your agent the power to perform every visual check imaginable, from capturing entire pages to pulling simple metadata points.

#	TOOL	DESCRIPTION
01	`take_element_screenshot`	Takes an image of one specific part of the page, defined by its CSS selector.
02	`get_page_metadata`	Retrieves basic site information like titles and page size without taking any pictures.
03	`generate_pdf`	Creates a professional PDF document from a specified web address.
04	`take_full_page_screenshot`	Captures a single image that includes every pixel from the top edge to the bottom edge of the website.
05	`take_screenshot`	Takes a standard screenshot of an entire URL.
06	`take_viewport_screenshot`	Captures a screenshot sized to exact dimensions, simulating a specific device view.

See It in Action

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

U Take a screenshot of https://vinkius.com.

I've captured the screenshot for you! You can access the high-quality image via the provided URL. Would you like a full-page capture or a PDF version as well?

U Capture the full page of https://github.com/vinkius.

Full-page capture initiated! I've retrieved the image showing the entire GitHub profile. It includes all repositories and contributions. Would you like the direct link?

U Show metadata for https://www.google.com.

I've retrieved the metadata for Google. The site title is 'Google' and the page size is approximately 50KB. Would you like me to take a screenshot now?

Frequently Asked Questions

01 **How does ScreenshotOne MCP handle full-page captures?**

The agent uses the `take_full_page_screenshot` tool to generate one single image that stretches from the very top of the page content to the bottom. You get a complete visual context, which is great for archiving entire articles.

02 **Can ScreenshotOne MCP capture just a specific section?**

Yes. If you only need the header or a contact form, use the `take_element_screenshot` tool. You specify the exact CSS element, and the agent captures nothing else, keeping your image clean.

03 **Is ScreenshotOne MCP better than just taking a regular picture?**

Absolutely. This MCP provides programmatic access to high-resolution web data, allowing you to capture specific viewports or metadata programmatically. It's not limited by your screen size.

04 **How do I get a PDF using ScreenshotOne MCP?**

You simply instruct the agent to 'Generate a PDF from this URL.' The `generate_pdf` tool handles the conversion, giving you a clean, structured document file ready for sharing.

05 **Does ScreenshotOne MCP work if I don't know what page size I need?**

You can use `get_page_metadata` first to understand the site's dimensions. Then, you can ask the agent to take a viewport screenshot using those retrieved metrics.

06 **How do I find my ScreenshotOne Access Key?**

Log in to your [**ScreenshotOne dashboard**](https://screenshotone.com/app/settings), and you will find your Access Key under the 'API Keys' section. Copy and paste it below.

07 **Can the agent capture elements by ID or Class?**

Yes. Use the `take_element_screenshot` tool providing a valid CSS selector (e.g., `#header` or `.product-card`). Your agent will return a screenshot of just that element.

08 **Is PDF generation supported for all sites?**

Yes. The `generate_pdf` tool works for any publicly accessible URL, converting the website content into a high-quality PDF document instantly.

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 `"screenshotone": { "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

ScreenshotOne 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 ScreenshotOne. 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	ScreenshotOne MCP
Server ID	`019d847b-831d-71c0-9c33-2e8e29521170`
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/screenshotone.