MCP SERVER | NO CODE | CLOUD HOSTED

Urlbox MCP

Generate flawless visual proofs from any website or code.

Urlbox lets you render live websites into high-quality PNGs, JPEGs, and PDFs using a cloud API. This MCP handles complex responsive layouts and dynamic content so your agent can generate exact visual proofs of any public URL or raw HTML code base in multiple formats.

A+ Quality Score 100/100

screenshot-api | pdf-generation | web-rendering | html-to-image | automation | headless-browser

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/urlbox-alternative

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.

Urlbox MCP

12 tools available
Cloud-hosted on Vinkius

Need to prove what a webpage looks like? This connection lets your AI client capture it visually. You feed it a URL, and the system returns flawless screenshots—PNGs, JPEGs, WebPs—no matter how complicated the site's layout is. If you need a full document backup, it handles converting entire pages to customized PDFs. Developers can also use this MCP to test raw code snippets by rendering HTML directly into images, or they can focus on just one area using CSS selectors. It’s about getting perfect visual assets without leaving your agent flow. Since Vinkius is the #1 catalog for these kinds of tools, you get access right here, letting your agent handle everything from checking usage stats to running complex renders in a single chat session.

Core Capabilities

01 — Capture Visual Proofs

Generate instant screenshots and image files (PNG, JPG, WebP) of any public URL.

02 — Convert Pages to Documents

Transform entire web pages into high-fidelity PDF documents with customizable page sizes.

03 — Isolate Specific Content

Use CSS selectors to capture only a specific element, like a chart or price table, from a live webpage.

04 — Test Raw Code Styles

Render raw HTML code directly into images for visual testing of styles or assets.

05 — Monitor Usage and Assets

Track credit usage, list active webhooks, or check storage buckets directly through your agent.

One Click on Vinkius — From Prompt to Execution

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

01 First, you subscribe to this MCP and enter your unique Urlbox API Key into your AI client's settings.

02 Next, you prompt your agent with the URL or HTML code you want rendered. The agent then uses the appropriate tool within the MCP to send the request to the cloud service.

03 Finally, your agent returns the captured visual asset—whether it's a PNG file, a PDF link, or status updates—ready for you to review.

The bottom line is, you get reliable web rendering capabilities integrated directly into your AI workflow.

Built For

This MCP is essential for QA Engineers who need visual proof of bugs, Frontend Developers testing responsive layouts, and Marketing teams needing consistent ad assets. If your job involves verifying what a live website looks like across different platforms or creating content back-ups, you'll use this.

QA Engineer

Uses the MCP to generate PNG screenshots of specific URLs and selectively capture elements (like forms or error messages) to prove a bug exists.

Frontend Developer

Renders raw HTML code using this MCP to visually test how different CSS snippets appear before committing them to the main build.

Digital Marketing Manager

Automates the process of creating visual proofs—like social media previews or ad mockups—by generating high-quality screenshots and PDFs for campaign assets.

What Changes When You Connect

01 Stop manual screenshots. Instead of taking a picture and hoping it captures everything, use the `capture_png` tool to guarantee you get a high-fidelity image capture of any URL.

02 Need legal proof that an article looked a certain way? Use `capture_pdf` to convert entire webpages into standardized PDF documents with customizable page sizes for reliable archiving.

03 Don't waste time on full-page captures when you only need one thing. The `render_selector` tool lets your agent focus and capture just the pricing table or a specific chart, making your assets clean and targeted.

04 Test frontend code without running a local server. Simply use `render_html` to turn raw HTML snippets into images so you can visually check styles in place.

05 Build robust pipelines by using asynchronous methods like `render_async`. This allows your agent to start a long render job now and process the result later via webhooks, without timing out.

Real-World Applications

Debugging Layouts for Clients

A QA engineer needs to show a client that a specific pricing widget is misaligned. Instead of manually cropping screenshots, they ask their agent to use `render_selector` on the live page and get a perfectly cropped image of just the offending widget.

Creating Content Backups

A content creator finishes an article and wants a permanent record. They prompt their agent to use `capture_pdf` on the URL, instantly generating a professional, A4-sized PDF backup of the entire piece.

Validating Marketing Mockups

A marketing team needs visual assets for an ad campaign. They instruct their agent to use `capture_png` on three different landing pages and gather all high-quality, consistent screenshots needed for the final creative brief.

Testing Isolated Components

A developer wants to check if a new JavaScript component (like a complex graph) renders correctly. They use `render_html` by pasting the raw code, and their agent returns an image so they don't have to run it in a full browser environment.

Patterns to Avoid

Trying to read complex JS client-side data

✗ AVOID

Asking the agent to 'extract all user names from the page object.' This fails because the AI can't execute JavaScript in a browser environment.

✓ INSTEAD

You must use visual capture tools. To get an image of the whole page, use `capture_png`. If you only want a specific section, run `render_selector` with the correct CSS identifier.

Handling large sites that time out

✗ AVOID

Asking the agent to capture a massive site like Wikipedia in one go. The request will fail or timeout because it's too much data for a synchronous call.

✓ INSTEAD

Use asynchronous rendering methods, specifically `render_async`. This starts the job and sends the results back later via webhooks, ensuring reliability for large captures.

Needing only one element from a page

✗ AVOID

Telling the agent to 'get me the price' when the page has many elements. The resulting screenshot will be messy and contain irrelevant data.

✓ INSTEAD

Be precise. Use `render_selector` by providing the exact CSS selector, like `#pricing-table`. This forces the capture tool to isolate only the needed element.

The Right Fit

Use this MCP if your goal is visual verification: you need a screenshot, a PDF copy, or an image of one specific part of a live website. You're confirming *how* something looks visually in its rendered state. Don't use it if you need to interact with the underlying data—for example, if you only need to list all available proxies without knowing what they look like, simply using `list_proxies` is enough. If your goal is pure data extraction (e.g., fetching a user ID or an email address), use standard database query tools instead of attempting to parse text from an image.

The struggle of visual proofing

Right now, proving how a site looks—especially if it's responsive or has dynamic content—is a manual nightmare. You open the URL, resize your browser window to check mobile views, take a screenshot, then repeat for desktop and tablet. If you need an archive copy of a blog post, you have to print it, export it as PDF, and hope the formatting doesn't break. It’s click-heavy, time-consuming, and prone to human error.

With this MCP, your agent handles all that messy work for you. You give it the URL, tell it the format (PDF or PNG), and it manages responsive layouts automatically. The result is a perfect visual asset, instantly ready for use—no manual resizing, no broken formatting.

Capture Web Assets with Urlbox

You don't have to open multiple tabs or switch between design tools just to verify a layout. Instead of copying text and pasting it into an image editor, you let the agent run `render_selector`. This command finds a specific element—like your main call-to-action button—and returns a perfect cropped PNG of *only* that element.

The biggest change is control. You move from general web browsing to targeted asset generation. The MCP gives you absolute certainty over the visual data, letting you build accurate proofs and archives in seconds.

Urlbox: Web Rendering & Capture (12 Tools)

These tools let you manage the entire lifecycle of a web asset—from checking account limits to capturing tiny CSS components or full-page PDFs.

#	TOOL	DESCRIPTION
01	`get_account_info`	Retrieves basic profile information about the connected Urlbox account.
02	`list_proxies`	Provides a list of available proxy locations that can be used for rendering web pages.
03	`get_render_status`	Checks the current status of an asynchronous rendering job you started.
04	`get_credit_usage`	Retrieves your current usage statistics and remaining credit balance.
05	`list_storage_buckets`	Lists all the storage buckets associated with your account for asset management.
06	`list_webhooks`	Retrieves a list of active webhooks configured on your account.
07	`render_async`	Starts an asynchronous rendering job for a URL, and the results are sent to a specified webhook endpoint.
08	`render_html`	Converts raw HTML code directly into a rendered image file.
09	`capture_pdf`	Takes an entire live URL and converts it into a comprehensive PDF document.
10	`capture_png`	Captures a standard PNG screenshot of an entire web page or specific area.
11	`render_selector`	Renders and captures only the content matching a specific CSS selector on a live webpage.
12	`render_sync`	Performs an immediate, synchronous rendering of a URL and returns the result directly.

See It in Action

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

U Take a high-quality screenshot of 'https://vinkius.com'.

Screenshot captured! I've rendered 'https://vinkius.com' as a high-quality PNG. You can view or download the image here: [Render URL]. Shall I convert this page to PDF as well?

U Convert the article at 'https://example.com/blog/1' to an A4 PDF.

Conversion successful! I've generated an A4 PDF for the provided URL. Here is your download link: [PDF URL]. Is there anything else you'd like to capture?

U Capture only the element '#pricing-table' from 'https://mysaas.com'.

Selective render complete! I've captured only the element matching '#pricing-table' from 'https://mysaas.com'. Here is the cropped image: [Element URL].

Frequently Asked Questions

01 **How does Urlbox handle complex websites like those with JavaScript?**

It handles dynamic content because it uses a cloud API that renders the page as if a real browser were viewing it. This means you get accurate screenshots even if the site runs heavy JavaScript.

02 **Can I capture only part of a webpage using Urlbox? What tool do I use?**

Yes, you can isolate elements with precision. Use the `render_selector` tool by providing a CSS selector (like '#footer') and it will capture just that specific section.

03 **What is the difference between `capture_png` and `capture_pdf` using Urlbox?**

`capture_png` gives you an image file for a single view, perfect for ads or proofs. `capture_pdf`, however, converts the whole page into a document format, ideal for reports and archives.

04 **Does Urlbox support large-scale rendering jobs?**

Yes. For very long URLs or complex renders that might time out, use `render_async`. This starts the job in the background and you can track its status using `get_render_status`.

05 **Is there a way to list my billing details with Urlbox MCP?**

You can check your usage stats by calling the `get_credit_usage` tool. This will show you how many credits you've used and what your current balance is.

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 `"urlbox-alternative": { "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

Urlbox 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 Urlbox. 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	Urlbox MCP
Server ID	`019dd17f-b3c9-7392-a072-9fe3d1cfb77a`
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/urlbox-alternative.