MCP SERVER | NO CODE | CLOUD HOSTED

# Plex MCP

## Know exactly what's playing or find anything in your library.

Plex connects your AI agent directly to your private media library. Search, manage, and track everything from movies and TV shows to photos and music without opening a single app. Check who's watching what right now, find detailed cast information for any title, or even mark items as watched using only natural conversation.

A+ Quality Score 100/100

media-server | streaming | library-management | playback-control | media-metadata | self-hosted

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

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.

# Plex MCP

15 tools available
Cloud-hosted on Vinkius

Tired of juggling multiple apps just to check your home media? This MCP connects your AI agent straight into your Plex Media Server, letting you manage your entire library through simple talk. You can ask it things like, 'Who's watching that detective show right now?' or 'What are the best-reviewed sci-fi movies I haven't seen?' It pulls all that data together instantly. Because Vinkius hosts this connector, you connect once from any compatible AI client and get immediate access to your entire media collection management system. You can ask it for detailed information on an actor's filmography or list every playlist you've ever made. This isn't just a search tool; it lets you monitor active streaming sessions and even tell the server, 'Mark this movie as watched.' It makes managing thousands of files feel like chatting with a smart librarian.

## Core Capabilities

**01 — View Server Health Status**

Check if the Plex Media Server is running correctly and get its current version information.

**02 — Monitor Active Streams**

See a real-time list of every user who is currently watching media, including their progress and device type.

**03 — Browse Media Collections**

List all major sections on your server, such as 'Movies,' 'TV Shows,' or 'Music' libraries.

**04 — Find Specific Content Details**

Pull comprehensive details about any item, including its cast, genre, runtime, and file path.

**05 — Manage Viewing Progress**

Update an item's status by marking it as watched or unwatched directly through conversation.

## One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to the Plex MCP and provide your specific Plex Token and Server URL.

02 Connect this MCP to your preferred agent via Vinkius, granting it access permissions for reading media data.

03 Ask your AI client a question like, 'Show me what's next in my watchlist,' and the tool returns the required content.

The bottom line is you talk to your AI agent, which talks to Plex, and you get an answer without ever opening the Plex app.

## Built For

This MCP is for anyone whose life revolves around media consumption or managing large digital archives. If you're frustrated by having to click through dozens of folders just to find out who watched what, this is for you.

**Home Media Enthusiast**

Uses the MCP to check which shows are currently playing in the house and finds new content to binge next.

**System Administrator**

Runs diagnostics on the server, checks connection health, and manages library sections for proper indexing.

**Family Manager**

Monitors what different family members are watching across various devices and helps organize shared playlists.

## What Changes When You Connect

01 Stop guessing what to watch. Use the `get_on_deck` tool to instantly see the next unwatched episode for any series you follow.

02 Keep track of every viewing habit with `get_sessions`. You immediately know who's watching and where they left off, which is great for family coordination.

03 Need proof that your server is working? Running `get_server_info` verifies connectivity and checks the overall health status in seconds.

04 Organize content by pulling specific item lists using `get_library_content`. You can filter results to only see movies or photos from a particular collection.

05 Manage history effortlessly. Use `mark_watched` when you finish an episode, keeping your 'On Deck' list accurate and reducing manual clicks.

06 Quickly find that obscure movie detail using `get_metadata`, getting the director, full cast, and summary without navigating deep menus.

## Real-World Applications

**Tracking a Mystery Viewing Session**

A family member asks their agent, 'What is currently playing in the living room?' The agent uses `get_sessions` to reply with real-time details, including the user's name and exact playback progress. No guesswork necessary.

**Building a Watchlist of New Content**

A user asks for all new sci-fi additions since last week. The agent uses `get_recently_added` combined with `search_library`, pulling titles and metadata instantly, saving hours of manual browsing.

**Debugging Library Issues**

The server admin runs into a new folder that isn't showing up. They use the agent to execute `refresh_library` on that specific section, fixing the indexing problem immediately.

**Curating a Themed Playlist**

Instead of manually finding and adding items, the user asks the agent to list all music from a certain era. The agent uses `get_playlists` followed by `get_library_content` to build a perfect playlist structure.

# Patterns to Avoid

### Asking for general content

✗ AVOID

A user asks, 'Tell me about the movies.' The agent responds with an unhelpful list of 50 generic titles because it doesn't know what section you mean.

✓ INSTEAD

Always specify the scope. To see all your movie sections, use `get_libraries`. If you only want to search one area, start by calling `get_library_content` with a specific key.

### Assuming status changes

✗ AVOID

A user says, 'I watched the show,' but doesn't specify which item. The agent fails because it needs context.

✓ INSTEAD

You must provide enough detail for the action. To mark a specific title as finished, you need to use `mark_watched` and identify the correct media item.

### Confusing search with browsing

✗ AVOID

A user asks, 'Show me my shows.' The agent might give them generic results instead of a structured list.

✓ INSTEAD

To get a comprehensive overview of your available sections (movies, photos, music), always start by using `get_libraries`.

## The Right Fit

Use this MCP if your primary need is managing, tracking, and retrieving data from a highly structured media collection. If you need to know who watched what, or pull metadata on specific cast members, this is the right tool. However, don't use it if you just want to browse files via drag-and-drop; that's a local file explorer job. Also, don't use it for external content management, like syncing with Netflix or Spotify; this MCP only talks to your private Plex server. If you need complex data transformation (like combining library data with user database records), you might need an advanced workflow tool instead.

## The Pain of Manual Media Tracking

Think about the last time a friend asked what movie you watched, or when you needed to check if your family had seen that new season. You usually have to open Plex, click through multiple tabs—Library, Sessions, Metadata—and then manually copy down the details or status updates.

With this MCP, those manual steps vanish. Your agent handles all that clicking and cross-referencing for you. Instead of navigating complex menus, you simply ask, 'What are we watching right now?' and you get a clear answer.

## Plex MCP: Instant Media Context

No more digging through folders or remembering specific IDs. You don't have to manually check server health or list out every library section just to get an overview.

The system gives you a single, conversational point of access to your entire media archive. It’s like having an expert librarian who knows exactly where everything is.

# Plex With 15 Tools

These tools give your AI agent the specific ability to read data about your server's libraries, check active streams, search media, and update viewing status.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `get_all_episodes` | Retrieves a complete list of episodes for any given television series. |
| 02 | `get_children` | Finds related content, such as getting all seasons from a show or all albums from an artist. |
| 03 | `get_libraries` | Lists every major section on your server, detailing what type of media it holds (movies, photos, etc.). |
| 04 | `get_library_content` | Filters and retrieves all content from a specific library you designate. |
| 05 | `get_metadata` | Gathers deep details on any item, including the director, cast list, summary, and original file path. |
| 06 | `get_on_deck` | Identifies the next unwatched episode or album you should watch to continue a series. |
| 07 | `get_playlist_items` | Pulls all the specific items, titles, and types contained within a playlist. |
| 08 | `get_playlists` | Lists every single playlist you have created on the server, along with their content count and type. |
| 09 | `get_recently_added` | Shows a list of all movies, albums, or episodes that were recently added to your media library. |
| 10 | `get_server_info` | Provides basic information and status checks for the entire Plex Media Server instance. |
| 11 | `get_sessions` | Lists all currently active viewing sessions, including who is watching what and their playback progress. |
| 12 | `mark_unwatched` | Resets the watch status of a media item so it will appear as unwatched again. |
| 13 | `mark_watched` | Updates an item's record to show that you have finished watching it, removing it from 'On Deck'. |
| 14 | `refresh_library` | Forces the server to re-scan a designated library after you've manually added or changed media files. |

| # | TOOL | DESCRIPTION |
|---|---|---|
| 15 | search_library | Searches across all your libraries for specific titles, episodes, or artists by name. |

# See It in Action

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

**U** Show me what's currently playing on my Plex server.

Active sessions: 1. User 'John' watching 'The Last of Us S01E03' at 34:15/58:22 (Roku, Direct Play). 2. User 'Sarah' watching 'Oppenheimer' at 1:45:30/3:00:15 (iPhone, Transcoding 1080p).

**U** Show me my movie libraries.

Found 3 libraries: 1. 'Movies' (key: 1, type: movie, 1,245 items) — 2. '4K Movies' (key: 5, type: movie, 342 items) — 3. 'Kids Movies' (key: 8, type: movie, 189 items).

**U** What's on deck for me to watch?

On Deck: 1. 'The Mandalorian S03E05' — 2. 'Stranger Things S04E07' — 3. 'Ted Lasso S02E03' — 4. 'The Office S07E19' — 5. 'Severance S01E04'. All are next unwatched episodes of shows you're currently watching.

# Frequently Asked Questions

01 **How does the Plex MCP know what content I own?**

The MCP connects directly to your private Plex Media Server via a secure token. It can only access data that resides on your server, so it's always using your local library information.

02 **Can the Plex MCP help me find my favorite movie details?**

Yes. Using `get_metadata`, you can get comprehensive reports on any item, including its cast list, genre, and runtime, all through natural conversation.

03 **Does the Plex MCP track who is watching what?**

Absolutely. The `get_sessions` tool provides a real-time dashboard showing every user currently streaming media, along with their progress and device type.

04 **How do I update my watch history using the Plex MCP?**

You use the dedicated status tools. Simply asking the agent to `mark_watched` or `mark_unwatched` handles the necessary database updates for you.

05 **Can the Plex MCP refresh my library if I add new files?**

Yes, after manually adding media, you can use the `refresh_library` tool. This forces a scan of that specific section so your AI agent sees the new content immediately.

# 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 `"plex": { "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

# Plex 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 Plex. 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 | Plex MCP |
| Server ID | `019d846e-29cb-720e-9c64-6335f05000be` |
| 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/plex.