MCP SERVER | NO CODE | CLOUD HOSTED

Health XML Export Parser MCP

Stop crashing your AI with massive health files.

The Health XML Export Parser handles huge Apple Health or Google Fit XML files safely. Stop giving your AI client massive, unmanageable data dumps that crash its context window. This MCP locally parses multi-megabyte health exports, aggregating millions of records (like step counts and heart rates) into structured summaries the AI can actually read.

A+ Quality Score 100/100

xml-parsing | health-data | data-processing | fitness-tracking | data-extraction

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/health-xml-export-parser

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Health XML Export Parser MCP

1 tools available
Cloud-hosted on Vinkius

Trying to feed an entire year's worth of biometric data—think hundreds of megabytes—into a chat interface is a recipe for failure. Your AI client will crash before it even gets past the raw text dump. This MCP solves that problem by running a high-performance parser locally on your machine. Instead of sending millions of individual lines to your agent, it intelligently analyzes the file's structure and aggregates the data. It tells the AI exactly what types of records exist—like StepCount or SleepAnalysis—and provides total counts along with safe samples for deeper inspection. This whole process keeps all sensitive health data private because it never leaves your local machine. When you connect this MCP through Vinkius, you get reliable access to powerful data processing without any context window limits.

Core Capabilities

01 — Analyze raw export files

You provide the file path and the tool parses massive Apple Health or Google Fit XML exports safely.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/health-xml-export-parser** — connect your AI agent in three steps.

01 Provide the absolute file path to your exported health data (e.g., export.xml).

02 The MCP uses a local, high-performance parser to ingest the massive XML file without overwhelming your AI client's context.

03 You receive an aggregated summary that tells your agent exactly which types of metrics exist and how many records there are.

The bottom line is you get structured, manageable data summaries from enormous health files, letting your AI client actually use the information instead of getting overloaded by it.

Built For

Data analysts and bio-tech researchers who routinely deal with large volumes of personal biometric data. If you're tired of manually sifting through massive XML files just to get a summary for your AI agent, this MCP is for you.

Bioinformatics Researcher

They feed the raw export file into the tool and then instruct their agent to calculate year-over-year trends in specific metrics like heart rate variability.

Digital Health Consultant

They use the MCP to quickly identify if a client's data is primarily sourced from one device or multiple sources, helping them diagnose data gaps.

Personal Data Analyst

They want to understand what types of metrics their fitness tracker captures by running the tool on an export and asking for a list of all available record types.

What Changes When You Connect

01 Handles multi-megabyte files without crashing. You can feed the tool an entire year of data, and your agent won't lose context on record count or structure.

02 Saves you from manual parsing. Instead of having to copy/paste millions of lines into a spreadsheet just for a summary, you let the MCP aggregate the raw numbers automatically.

03 Maintains total privacy. Since this process happens entirely locally on your machine, sensitive health data never leaves your device or is exposed to external services.

04 Provides immediate structural context. You don't just get a blob of text; the tool tells your agent exactly what kinds of metrics exist (like SleepAnalysis) and how many records there are.

05 Makes advanced analysis possible. You can ask your AI client complex questions about data structure, like determining which devices were responsible for the bulk of step counts.

Real-World Applications

Figuring out overall metrics captured

A user wants to know if their Apple Health export includes sleep stage details. They run the `parse_health_export` tool and ask their agent, 'What types of records are present?' The agent immediately replies with a list of all available metric types (e.g., HeartRate, SleepAnalysis), giving them an instant inventory.

Comparing data sources

A consultant needs to know if client data comes from the watch or the phone. They use `parse_health_export` on a mixed export and ask their agent to summarize the primary device identifiers, allowing them to quickly diagnose potential syncing issues.

Setting up long-term tracking reports

A researcher needs to build a longitudinal study. Instead of manually processing 10 different years of XML files, they run `parse_health_export` on each one and have their agent aggregate the total counts for core metrics across all datasets.

Testing data completeness

A user suspects a gap in their recorded activity. They use `parse_health_export` and ask their agent to list all record types, immediately seeing that 'RunningDistance' is zero, confirming they forgot to sync certain activities.

Patterns to Avoid

Pasting the raw XML

✗ AVOID

A user sees a massive block of XML code and thinks they can just paste it into their agent's prompt, hoping it will work.

✓ INSTEAD

Don't paste anything. You must use the `parse_health_export` tool by providing the file path. This runs the data through a secure parser first, which structures the raw text so your AI client can process it safely.

Asking for interpretation of unstructured data

✗ AVOID

The user feeds in an XML and asks, 'Tell me about my health.' The agent gets overwhelmed by the sheer volume of tags and numbers and gives a vague, unhelpful answer.

✓ INSTEAD

Let the `parse_health_export` tool run first. It aggregates the data into a summary that provides clear metrics (e.g., 1M StepCounts), giving your AI client concrete numbers to work with.

Relying on memory or manual summaries

✗ AVOID

The user tries to summarize months of activity by reading printed reports and manually writing the key metrics into a chat window.

✓ INSTEAD

Simply point your agent at the file using `parse_health_export`. The tool handles the heavy lifting, providing the clean, aggregated data points you need instantly.

The Right Fit

Use this MCP if your primary goal is to extract high-level metrics and structural information from massive, raw health XML files. You need a summary—a count of record types (like StepCount or HeartRate) and totals—without having the AI client crash on context overflow. Don't use it if you are trying to modify the data itself (you can't edit the source file through this MCP). Also, don't use it if your health data is already clean, structured JSON; in that case, a standard data connector is better. This tool exists only for messy, raw XML exports from platforms like Apple Health.

The pain of raw biometrics data

Right now, if you want your AI agent to analyze your health history—say, tracking sleep patterns over a year—you have to deal with XML files that are huge. You download the export, and when you try to give it to Claude or Cursor, the sheer volume of tags and numbers crashes the session immediately. You're left staring at an error message because the raw data is too massive for any chat window.

With this MCP, you point your agent at the file path using `parse_health_export`. The tool runs locally, parsing out the noise and giving your AI client a clean summary: 'This dataset has 1.2 million StepCounts, 500k HeartRates.' Suddenly, that multi-megabyte problem is solved, and you're talking to an agent that knows exactly what data it's working with.

Get actionable insights using parse_health_export

Manual analysis requires opening the file in specialized software, manually filtering by date range or metric type, and then copy-pasting small, digestible chunks of data into a separate spreadsheet. This process is time-consuming, error-prone, and doesn't help your AI agent at all.

By using `parse_health_export`, the entire workflow changes. You get immediate, actionable summaries of record types and totals in one step. Your agent moves from being overwhelmed by data to actively analyzing it.

Health XML Export Parser MCP (1 Tool)

These tools allow you to parse and summarize massive, complex health data export files without overwhelming your AI client's context window.

#	TOOL	DESCRIPTION
01	`parse_health_export`	Provide the file path to parse Apple Health or Google Fit XML exports, aggregating data to prevent context overflow.

See It in Action

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

U Scan my Apple Health export.xml and tell me what types of metrics are tracked.

Your export contains exactly 1,240,000 StepCounts, 500,123 HeartRates, and 450 SleepAnalysis records. Which one should we analyze first?

U Look at my health export and summarize my device sources.

The data is primarily sourced from an 'Apple Watch Series 8' and an 'iPhone 14 Pro'.

U Based on the sample data of 'SleepAnalysis', what format does Apple use to track sleep?

Apple tracks sleep using a StartDate, EndDate, and a 'value' string representing the sleep state (e.g., Core, REM, Deep).

Frequently Asked Questions

01 **Can I use Health XML Export Parser for Google Fit files?**

Yes, this MCP is designed to handle both Apple Health and Google Fit XML exports. You simply provide the file path, and the parser handles the specific structure of either export type.

02 **Does using parse_health_export affect my data privacy?**

No. The parsing happens entirely on your local machine. Your sensitive health metrics are never uploaded or stored anywhere outside of your controlled environment.

03 **What is the maximum size file that parse_health_export can handle?**

It is built for multi-megabyte files, designed specifically to prevent context window overflow. It aggregates data intelligently rather than transmitting the full raw text.

04 **Does this MCP help me analyze step counts or heart rates?**

Yes, it first identifies that those metrics exist and provides their total count. Your agent then uses that structured summary to run detailed analysis on them.

05 **Do I need to write any code to use Health XML Export Parser?**

No. You just interact with the tool by providing the file path, and your AI client handles the rest of the data processing automatically.

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 `"health-xml-export-parser": { "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

Health XML Export Parser 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 Health XML Export Parser. 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	Health XML Export Parser MCP
Server ID	`019e38a6-ed42-71e0-a6df-fe3b585a4b1f`
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/health-xml-export-parser.