MCP SERVER | NO CODE | CLOUD HOSTED

Oura MCP

Analyze Sleep, Activity, and Recovery Data Instantly

Oura MCP connects your Oura Ring health data directly into any AI agent. You can ask questions about sleep scores, activity levels, readiness metrics, heart rate variability (HRV), and workout history using natural conversation. It turns complex biometric logs into simple, actionable insights for biohackers and athletes.

A+ Quality Score 100/100

wearable-tech | sleep-tracking | health-metrics | hrv | activity-monitoring

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

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.

Oura MCP

7 tools available
Cloud-hosted on Vinkius

Connecting your Oura Ring to this MCP lets you analyze all your personal health data without logging into a separate app. You simply ask your AI agent questions—like 'How was my sleep last night?' or 'What's my recovery status today?' Your agent pulls the necessary metrics, such as deep/REM sleep cycles, daily step counts, and resting heart rate, and gives you a clear answer. This is huge for people who need to correlate their mood tags with their actual biometric performance. Instead of sifting through multiple dashboards, all this data lives together in one place on Vinkius, accessible by your preferred AI client.

Whether you’re an athlete monitoring recovery or just trying to understand why you feel tired, this MCP lets you get a full picture of how sleep impacts readiness and what specific workouts are hitting your targets. It's about getting the narrative out of the raw numbers.

Core Capabilities

01 — Summarizing Sleep Metrics

Retrieve detailed analysis on sleep scores, stages (deep, REM, light), efficiency, and how long it took you to fall asleep.

02 — Tracking Daily Activity Progress

Get metrics like steps taken, calories burned, MET minutes logged, and daily activity goal achievement percentages.

03 — Assessing Recovery Status

Monitor your readiness score alongside key physiological data points, including HRV, resting heart rate, and body temperature readings.

04 — Reviewing Workout History

Browse structured workout logs, detailing the activity type, duration, calories burned, and specific heart rate zones reached during exercise.

05 — Correlating Behavior Tags

Access user-entered tags for mood, energy levels, or substances to correlate with your sleep quality and readiness scores.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your Oura Personal Access Token.

02 Your AI client connects and authenticates the data stream from your wearable device.

03 You ask a question—for example, 'Compare my activity last week vs. this week'—and your agent processes the relevant metrics instantly.

The bottom line is that you talk to the data instead of navigating complex web interfaces.

Built For

This MCP is for anyone whose routine involves tracking physical performance and correlating it with recovery. It targets biohackers, dedicated endurance athletes, and general health enthusiasts who need more than just a daily score.

Endurance Athlete

Uses this MCP to check readiness scores before a long run, ensuring their HRV hasn't dropped too low, or checking workout logs to optimize training intensity.

Biohacker / Health Enthusiast

Asks the agent to correlate poor sleep quality (from get_sleep) with specific mood tags (get_tags) over a month to find behavioral patterns.

Physical Therapist or Coach

Retrieves historical activity data and heart rate metrics to track patient recovery rates between therapy sessions, providing objective proof of improvement.

What Changes When You Connect

01 You don't have to manually cross-reference data. By checking your readiness score using get_readiness, you immediately know if it’s safe to push hard or if recovery is needed.

02 Get a holistic view of performance by calling get_sessions; this single tool pulls together sleep metrics, activity totals, and overall health indicators for easy comparison.

03 Deep dive into your nocturnal habits. The get_sleep function breaks down exactly how long you spent in deep versus REM sleep, helping identify poor sleep patterns.

04 Better pattern recognition comes from linking data. You can use get_tags to correlate a low readiness score with a specific mood entry or behavior logged that day.

05 Workout analysis is simplified. Use get_workouts to review past exercises and track how your heart rate zones are changing over time.

Real-World Applications

Diagnosing Performance Slumps

A user notices a dip in performance. Instead of guessing, they ask their agent to compare get_sleep logs from the week before and after the slump, cross-referencing it with any mood tags (get_tags) entered during that period.

Optimizing Training Load

An athlete wants to know if they're overtraining. They ask their agent to fetch get_heart_rate data and compare the average resting heart rate against their historical baseline, guiding whether a rest day is necessary.

Understanding Activity Goals

A user wants accountability for movement. They prompt the system to check get_activity data for the last 30 days, getting a clear breakdown of average steps and calories burned versus their set goals.

Comparing Workout Types

Someone who tries different sports wants objective proof. They ask the agent to use get_workouts to pull data on both swimming and cycling sessions, allowing them to compare duration, calorie output, and average heart rate zones.

Patterns to Avoid

Checking metrics in silos

✗ AVOID

A user checks their sleep score in one app, steps in another, and HRV on a third. They spend 15 minutes copying numbers into a spreadsheet just to draw a rough conclusion.

✓ INSTEAD

Instead, ask your agent to use get_sessions or get_sleep together. The MCP pulls all necessary data points—like sleep score alongside activity totals—and presents them in a single narrative response.

Ignoring context

✗ AVOID

A user only looks at their readiness score today and decides to work out, ignoring the fact that they logged poor energy levels (get_tags) yesterday.

✓ INSTEAD

Always ask your agent to correlate get_readiness data with get_tags. This forces a check of behavioral context against your physiological metrics.

Asking vague questions

✗ AVOID

Typing 'How was my health?' into the prompt, which results in an unhelpful stream of raw numbers and unclear dates.

✓ INSTEAD

Be specific. Use get_workouts to ask, 'Compare my cycling workouts from last month.' This directs the agent to use the precise data needed for a concrete answer.

The Right Fit

Use this MCP if you need to understand the *relationship* between different types of health data. For example, if your sleep quality (get_sleep) seems low and you want to know if it impacts your daily energy levels (get_tags), this is perfect. You're not just looking for a number; you're building a narrative about your body's performance. Don't use this if you simply need to view raw data—for that, the Oura app itself works fine. However, if you want an AI agent to read those raw numbers and explain *what they mean* in the context of your goals or training plan, this MCP is necessary. It turns a collection of logs into actionable knowledge.

The Daily Data Dump Problem

Right now, tracking performance means logging into five different places: the sleep tracker for your nightly score, the activity app for steps, and maybe a separate mood journal. You spend time copying numbers, opening comparison charts, and manually asking yourself if those low-score days were due to poor sleep or just bad diet.

With this MCP, you ask one question—'Why was my recovery so low last week?' Your agent uses the Oura connection to pull data from get_sleep, get_tags, and get_readiness all at once. You stop managing dashboards; you start getting clear answers.

Oura MCP: Contextualizing Your Health Metrics

Manual analysis usually involves running a report on your get_heart_rate data, then manually finding the corresponding day in your get_tags history to see if stress levels match the spike. It's slow, and you often miss subtle correlations.

This MCP automates that correlation. You ask for the connection between your HRV spikes and your mood entries, and the agent handles the heavy lifting instantly. The data talks to itself.

Oura MCP with 7 Tools

Use these specific tools to pull highly detailed data points—like heart rate samples or workout logs—into your agent for deeper analysis.

#	TOOL	DESCRIPTION
01	`get_activity`	Pulls your daily step count, calorie burn, and activity goal completion status for a selected date range.
02	`get_heart_rate`	Retrieves detailed heart rate samples and HRV readings; remember to use narrow date ranges because the data is high-frequency.
03	`get_readiness`	Gets your daily readiness score, along with related body metrics like resting heart rate and temperature readings.
04	`get_sessions`	Retrieves a comprehensive summary of scores, including sleep data, activity totals, and overall recovery indicators for a date range.
05	`get_sleep`	Provides detailed logs on your night's sleep, covering total duration, deep/REM stages, efficiency, and wake times.
06	`get_tags`	Accesses all manually entered data points you logged about mood, energy, or behaviors for a given time period.
07	`get_workouts`	Lists both manual and auto-detected workouts, detailing the type of exercise and how long it lasted.

See It in Action

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

U How did I sleep last night?

Last night: Sleep score 85/100. Total: 7h 42m. Deep: 2h 10m, REM: 1h 45m, Light: 3h 15m, Awake: 32m. Efficiency: 92%. Onset: 15min. Restlessness: Low.

U What is my readiness score today?

Today's readiness: 82/100 (Good). HRV: 65ms, Resting HR: 51 bpm, Body temp: 36.6°C. Sleep balance: 8h 5min. Activity balance: Well recovered. Previous day strain: 14.

U Show me my activity from yesterday.

Yesterday's activity: Score 75/100. Steps: 8,450. Calories: 2,340 (active: 520). MET minutes: 145. Average HR: 72 bpm. Inactivity: 11h 20m. Daily goal: 85% achieved.

Frequently Asked Questions

01 **How do I use get_sleep with Oura MCP?**

You prompt your agent by asking a question about sleep cycles or efficiency for a specific date range. It will pull the data from get_sleep and summarize deep, REM, and light stages for you.

02 **Does Oura MCP track my heart rate in real time?**

No, it retrieves historical, high-frequency measurements using get_heart_rate. The data is sampled over specific intervals, so always use narrow date ranges when querying this tool.

03 **Can I correlate mood tags with my readiness score?**

Yes, you can ask the agent to cross-reference your get_tags entries (mood/energy) against your daily readiness data from get_readiness to spot behavioral patterns.

04 **Which tool should I use for total activity? Is it get_activity or get_sessions?**

If you want a comprehensive summary including sleep metrics and overall indicators, use get_sessions. If you only need the raw step count and calorie burn, use get_activity.

05 **Does Oura MCP handle workout data for different sports?**

Yes, it pulls workout data using get_workouts, which supports various types of logged or auto-detected activities, including duration, calories, and heart rate zones.

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 `"oura": { "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

Oura 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 Oura. 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	Oura MCP
Server ID	`019d8469-0f32-7178-9d05-96795c054bb4`
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/oura.