MCP SERVER | NO CODE | CLOUD HOSTED

WakaTime (Coding Stats) MCP

Know exactly how long you spent on every line of code.

WakaTime (Coding Stats) MCP connects your coding data to your agent. Get detailed stats showing total time spent across specific date ranges, track progress against custom goals, and list every project your IDE has recorded activity for.

A+ Quality Score 100/100

coding-stats | time-tracking | developer-productivity | wakatime | activity-monitoring

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/wakatime-coding-stats

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.

WakaTime (Coding Stats) MCP

14 tools available
Cloud-hosted on Vinkius

Need to know where you actually spend your development hours? This MCP analyzes the raw data from your IDE, giving you a clear picture of your coding habits without manual effort. You can ask your agent for summarized statistics—say, total time logged over the last 30 days—or track how far along you are with specific personal goals. It also lists every project your plugins detect, providing daily activity totals per workspace. If your work involves external apps like Google Calendar, it even pulls that data in so nothing slips through the cracks. Connecting WakaTime to Vinkius lets your agent use this MCP and synthesize all that complex time logging into a simple conversation. It’s pure insight: you get an objective view of who's working on what, right when you need to know.

Core Capabilities

01 — Calculate overall coding time

Retrieve total coding hours logged since your WakaTime account was created.

02 — Get period-specific stats

Fetch summarized coding statistics for custom date ranges like the last 7 or 30 days.

03 — List and check goals

See a list of your defined coding targets and review your current percentage completion for any goal.

04 — Track project activity

Get a comprehensive list of every detected project, along with daily time summaries for each one.

05 — Analyze raw keystrokes

Inspect raw coding pings and duration blocks to understand exact start times and gaps in your work sessions.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/wakatime-coding-stats** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your unique WakaTime API key.

02 Your agent connects the credentials and prepares access to your coding history.

03 You ask a natural language question—like 'How long did I work on the backend last month?'—and get an immediate, summarized answer.

The bottom line is you talk to your agent about your code time, and it pulls all the detailed data for you.

Built For

This MCP is essential for developers who need objective proof of their effort or team leads tired of chasing manual progress reports. It cuts through guesswork and gives hard numbers on where time actually goes.

Software Developer

Uses this to track personal productivity, identifying which specific projects consume the bulk of their coding time.

Team Lead/Engineering Manager

Monitors high-level project activity and goal progress across multiple team members without having to collect reports manually.

Technical Consultant

Analyzes coding patterns and historical heartbeats to recommend process optimizations or skill focus areas for a client.

What Changes When You Connect

01 Pinpoint wasted time. Use 'get_stats' to check specific date ranges and see if your coding hours align with where you thought they did.

02 Manage expectations. Check progress against custom targets using the 'list_goals' tool so team leads can report on goals, not just effort.

03 Visualize your work flow. The 'list_projects' function gives an immediate overview of all tracked projects and how much time was spent in each one.

04 Account for everything. If you track non-coding tasks like meetings in Google Calendar, the 'list_external_durations' tool makes sure that time is included.

05 Analyze the nitty-gritty. Need to know exactly when your agent saw you stop typing? The 'list_heartbeats' tool gives raw ping data for deep analysis.

Real-World Applications

The Scope Creep Audit

A team lead asks their agent, 'Show me the total time spent on Project Alpha versus Project Beta last quarter.' The agent runs the query using 'get_stats' and delivers a clear comparison, instantly identifying where focus drifted.

The Performance Review Prep

A developer needs to prove they hit their quarterly goals. They ask for their progress against all targets using 'list_goals', getting a comprehensive report that confirms 95% completion on their main objective.

Debugging Time Loss

A consultant suspects the team is losing time between coding sessions. They run the raw activity data via 'list_heartbeats' and review the gaps, pinpointing exactly where the workflow breaks down.

Cross-Functional Reporting

A developer needs to include meeting hours in their productivity report. Instead of manually logging it, they let the agent use 'list_external_durations' to pull verified time from their calendar.

Patterns to Avoid

Assuming activity is constant

✗ AVOID

Thinking that because you wrote a lot of code last week, your agent will automatically know the hours. You might just ask for 'last week's stats,' which is too vague.

✓ INSTEAD

Be specific with date ranges. Use the 'get_stats' tool and tell your agent exactly what dates to cover (e.g., 'from 2024-10-01 to 2024-10-31') for accurate results.

Ignoring non-coding time

✗ AVOID

Running a simple query that only looks at keystrokes, which completely misses the hours you spent in related tools like Jira or calendar meetings.

✓ INSTEAD

Always include external data. Use 'list_external_durations' to pull verified activity from apps outside of your IDE.

Overlooking project boundaries

✗ AVOID

Asking for a general total time without filtering by specific projects, resulting in one massive number that tells you nothing about focus.

✓ INSTEAD

Use 'list_projects' first to see all options, then ask the agent to summarize stats for just the most relevant area.

The Right Fit

Use this MCP if your primary need is objective, quantitative proof of effort. If you are a developer trying to justify time spent or track granular progress against measurable goals, this is it. You're looking at hard metrics: hours logged per project, total duration in specific date windows, and goal completion percentages.

Don't use this if you need qualitative feedback (e.g., 'Was the code clean?' or 'Is this feature cool?'). This MCP only handles time tracking data. If your problem is coordinating people or brainstorming solutions, a communication tool or a different type of agent will work better. You should use this when you want to measure *what* was done and *for how long*, not *how good* it was.

Tracking developer time used to feel like cross-referencing spreadsheets.

Right now, if a manager wants to know where the team spent their last month, someone has to manually go through dashboards, export CSVs for each project, and then spend hours compiling those into one readable report. It's tedious work that introduces human error every single time.

With this MCP, you tell your agent what you need—for example, 'Show me the total effort on Project X versus Project Y last quarter.' The agent accesses all the underlying data streams and delivers a clean, summarized comparison instantly. You get actionable insight without touching a spreadsheet.

Get WakaTime (Coding Stats) MCP to list project time & goals

The need to manually compile daily activity logs and reconcile external meeting times with core coding hours disappears. You don't have to correlate data between your IDE plugins, calendar apps, and task trackers anymore.

Your agent reads the complexity out of the process. It turns scattered raw keystrokes into structured, reportable intelligence at a single prompt.

WakaTime (Coding Stats): 14 Tools

These tools allow your agent to access every facet of your coding history—from raw heartbeats to high-level goal tracking.

#	TOOL	DESCRIPTION
01	`create_external_duration`	Records activity time that came from outside your IDE, like a calendar entry.
02	`create_heartbeat`	Sends a raw ping to record immediate coding activity when you start working.
03	`get_all_time_since_today`	Retrieves the total amount of time logged across all projects since your account was created.
04	`get_goal`	Pulls specific details about a user-defined coding goal you set up.
05	`get_stats`	Gathers summarized coding statistics for any given date range.
06	`list_commits`	Lists project commits and shows exactly how much time you spent on each one.
07	`list_custom_rules`	Shows you the rules currently set up to modify or remove coding activity records.
08	`list_durations`	Retrieves blocks of coding time that were joined together after a pause in typing.
09	`list_external_durations`	Gets activity time that was imported from external applications, such as Google Calendar.
10	`list_goals`	Displays a list of all coding goals you have set and how close you are to hitting them.
11	`list_heartbeats`	Retrieves the raw, ping-level data showing every instance you started coding in your IDE.
12	`list_projects`	Lists every unique project that has generated activity on your system.
13	`list_summaries`	Provides daily summaries of all recorded coding activity.
14	`update_custom_rules`	Allows you to change the rules that modify or delete your coding activity data based on patterns.

See It in Action

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

U Show me my coding stats for the last 7 days.

I've retrieved your stats for the last 7 days. You coded for a total of 24 hours and 15 minutes. Your top language was TypeScript (12h), followed by Python (8h). Your most active project was 'vurb-mcp-server'.

U What are my current coding goals and how am I doing?

You have 2 active goals: 'Code 4 hours daily' which is currently at 85% completion for this week, and 'Contribute to Open Source' which is at 40%. Keep it up!

U Give me a summary of my coding activity for yesterday.

Yesterday, you logged 5 hours and 30 minutes of coding activity. You worked on 3 projects: 'API-Gateway' (3h), 'Frontend-App' (2h), and 'Documentation' (30m).

Frequently Asked Questions

01 **How do I use WakaTime (Coding Stats) MCP to check my total time?**

You ask your agent for the 'get_all_time_since_today' statistic. It provides one single number representing every minute logged since you first connected your account.

02 **Can I track external work hours with WakaTime (Coding Stats) MCP?**

Yes, the 'list_external_durations' tool pulls activity from sources like Google Calendar. This ensures non-coding time is included in your overall picture.

03 **What does list_heartbeats do with WakaTime (Coding Stats) MCP?**

The 'list_heartbeats' tool retrieves the raw, minute-by-minute ping data from your IDE plugins. This is useful for forensic analysis of when work activity truly began and ended.

04 **How do I compare projects using WakaTime (Coding Stats) MCP?**

You can use the 'list_projects' tool to see all options, then ask your agent to run a summary query against those specific project names. This gives you clear time allocations.

05 **Can I update my coding rules with WakaTime (Coding Stats) MCP?**

Yes, the 'update_custom_rules' tool lets you adjust how WakaTime processes your activity data. You can modify patterns to ensure only relevant time is counted.

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 `"wakatime-coding-stats": { "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

WakaTime (Coding Stats) 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 WakaTime (Coding Stats). 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	WakaTime (Coding Stats) MCP
Server ID	`019e3908-2156-728b-85e9-d1f72c6a6dc6`
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/wakatime-coding-stats.

© 2026 Vinkius · vinkius.com · support@vinkius.com