MCP SERVER | NO CODE | CLOUD HOSTED

World Time (Keyless) MCP

Never manually calculate time differences again.

World Time (Keyless) gives your AI client precise, synchronized time data for any global timezone. It tells you what time it is right now—down to the millisecond—for locations like 'Europe/London' or based on your network IP address. You don't need an API key; just ask.

A+ Quality Score 100/100

atomic-time | iana-timezone | utc-offset | dst | synchronization | api-service

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/world-time-keyless

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.

World Time (Keyless) MCP

2 tools available
Cloud-hosted on Vinkius

Dealing with time zones and Daylight Saving Time (DST) math sucks. You shouldn't have to look up UTC offsets in a separate spreadsheet every time you plan a meeting or write code that needs to track global events. This MCP lets your agent access synchronized, atomic clock data across the entire world's IANA region list through natural conversation. It reads the exact current time for any zone and verifies its offset status instantly.

When you connect World Time (Keyless) via Vinkius, your AI client becomes a global master clock. You can ask it to find the precise local time anywhere on Earth or even check if a region is currently observing DST. It gives reliable JSON data including Unix timestamps and day numbers, which means your agent can handle complex scheduling tasks without you lifting a finger. Whether you're developing backend logic or just trying to figure out what time lunch is in Tokyo, this MCP handles the complexity so you don't have to.

Core Capabilities

01 — Check current global time

Get the precise local time and UTC offset for any named timezone, or automatically detect time using your network IP.

02 — Verify DST status

Determine if a specified location is currently observing Daylight Saving Time (DST) and what its official offset is.

03 — List all supported zones

Retrieve the full, valid list of IANA timezone strings to ensure you use the correct identifier in your requests.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/world-time-keyless** — connect your AI agent in three steps.

01 First, connect this MCP through Vinkius. No API key is required.

02 Next, prompt your AI client with a question about time, like 'What's the time in Paris?' or 'Check my local IP time.'

03 Finally, your agent receives structured data detailing the exact current time, the UTC offset, and whether DST applies to that location.

The bottom line is that you just talk about time zones, and the MCP handles all the complex calculation and lookup for you.

Built For

This connector is essential for anyone whose job involves coordinating across multiple geographic locations. If your pain point is manually cross-referencing clocks or figuring out time zone offsets before a meeting, this MCP solves it.

Software Developer

Uses the MCP to verify UTC offsets and synchronized time data for server logging, ensuring applications work correctly regardless of where they run.

Project Manager

Quickly checks local times and DST status for distributed team members before scheduling cross-continent meetings or deadlines.

Operations Analyst

Monitors time zone transitions, verifies week numbers, and confirms regional offsets for business reporting across different departments.

What Changes When You Connect

01 Eliminate time zone guesswork. By using the `get_current_time` tool, you get precise local times and UTC offsets for any location instantly. No more checking multiple websites to confirm a date.

02 Automate global scheduling checks. Need to plan a meeting spanning New York and Sydney? Ask your agent, and it provides the current time and DST status for both regions in one go.

03 Build robust codebases. Developers use this MCP to reliably verify UTC offsets and synchronized timestamps, which is critical for accurate server-side logging and event tracking.

04 Discover required identifiers easily. Use `list_available_timezones` when you're unsure of the correct IANA string (like 'America/Los_Angeles'), ensuring your subsequent time checks are always valid.

05 Handle IP inference automatically. You can ask for the current local time based on your network IP address, perfect for internal tools that need to know the user's immediate location.

Real-World Applications

Coordinating a Global Product Launch

A Project Manager needs to schedule a launch announcement across London, Mumbai, and São Paulo. Instead of consulting a time zone chart, they ask their agent: 'What's the local time in these three places?' The MCP immediately returns the current date, offset, and DST status for all three zones.

Debugging Time-Sensitive Code

A Developer needs to write a script that logs events based on precise timestamps. They use `get_current_time` with specific time zone identifiers in their prompt, confirming the exact offset and ensuring the timestamp is accurate for production logging.

Writing Time-Aware Documentation

An Operations Analyst needs to document a scheduled system maintenance window that happens during DST transition. They ask the agent to check 'Europe/London' for its current DST status, guaranteeing they use the correct timezone rules in their documentation.

Checking Local Meeting Times

A user is traveling and needs to know what time it is right now where they are. They simply ask their agent: 'What's the current time based on my IP?' The MCP provides a quick, accurate local reading without needing manual input.

Patterns to Avoid

Manual Time Zone Conversion

✗ AVOID

Trying to figure out if London is observing DST right now by cross-referencing multiple online calculators and Google searches. This is slow, error-prone, and outdated.

✓ INSTEAD

Use the `get_current_time` tool. Simply ask your agent for the time in 'Europe/London' and it instantly reports the current offset and if DST is active.

Assuming Standard Time

✗ AVOID

Writing code that calculates offsets based only on standard UTC rules, failing when a location switches to Daylight Saving Time.

✓ INSTEAD

Always use this MCP. It provides the dynamic data needed for time synchronization, confirming both the offset and DST state automatically.

Using general date functions

✗ AVOID

Relying on basic system clocks or database entries that don't account for IANA standards or complex regional rules.

✓ INSTEAD

Use `get_current_time` with a specific IANA timezone string (e.g., 'Asia/Tokyo'). This guarantees the data respects global timekeeping standards.

The Right Fit

Use this MCP if your core task involves synchronizing or referencing accurate, real-world clock times across multiple geographic regions. You need to know what time it is in London *right now*, not just when it was last updated.

Don't use this if you only need to convert a fixed date (e.g., 'What day of the week was 2015-06-15?'). For static historical lookups, other specialized calendar tools might work better. However, if your problem is: 'Given Location X and Location Y, what time difference exists right now, considering DST?' then this MCP is exactly what you need.

Always use the specific timezone identifier (e.g., `America/New_York`) instead of a simple city name when calling `get_current_time`. This prevents ambiguity and gives you reliable results.

The global clock problem is exhausting.

Right now, planning anything that touches different time zones feels like detective work. You open one tab for the client's location, another for your team's office hours, and a third to check if they are currently observing Daylight Saving Time. Then you copy-paste the data into a spreadsheet just to confirm a meeting slot.

With this MCP, that whole process disappears. You ask your agent about global time coordination, and it doesn't just give you an answer; it gives you structured proof: the exact current time, the offset, and confirmation of DST status for every location you care about.

Get precise time using World Time (Keyless) MCP

You stop manually checking if a region is on standard time or DST. You don't need to remember the correct IANA identifier for every city; you can ask your agent, and it handles the lookup.

The result is immediate and reliable. Your AI client doesn't just guess at a time—it pulls real-time, atomic clock data directly into your workflow.

World Time (Keyless) with 2 Tools

These tools let your agent check the current global time, list all supported IANA zones, and calculate complex UTC offsets.

#	TOOL	DESCRIPTION
01	`get_current_time`	Retrieves the exact current atomic time, UTC offset, and DST state for a specified timezone or based on your IP address.
02	`list_available_timezones`	Provides a comprehensive list of all valid IANA timezone regions supported by the service.

See It in Action

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

U What time is it in Tokyo right now?

In Tokyo (Asia/Tokyo), it is currently Tuesday, March 31, 2026, at 09:15 PM. The UTC offset is +09:00 and they are not currently observing Daylight Saving Time.

U Check the current time based on my IP.

Based on your network IP, your local time is 10:30 AM. Your timezone is America/Sao_Paulo (UTC -03:00).

U Is London currently in Daylight Saving Time?

Checking Europe/London... Yes, London is currently observing Daylight Saving Time (BST). The offset is UTC +01:00. The transition back to GMT will occur later this year.

Frequently Asked Questions

01 **How do I check the current time using World Time (Keyless) MCP?**

You simply ask your agent for the current time in a specific timezone. You can provide an IANA string like 'Asia/Tokyo' or ask it to infer based on your network IP.

02 **Do I need an API key for World Time (Keyless) MCP?**

No, you don't. This MCP is designed without the friction of manual signups or complex keys, making it easy to connect and use immediately.

03 **What if I want to list all supported timezones using World Time (Keyless) MCP?**

You call the `list_available_timezones` tool. This returns a comprehensive list of all valid timezone identifiers, helping you write more precise prompts later.

04 **Can this MCP handle Daylight Saving Time transitions?**

Yes, absolutely. The service checks and reports whether the location is currently observing DST and provides the correct offset for that specific moment in time.

05 **Does World Time (Keyless) MCP work if I don't know the exact timezone string?**

It can guess based on your network IP address, giving you a local time reading. For maximum accuracy, it's best to use a specific IANA identifier.

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 `"world-time-keyless": { "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

World Time (Keyless) 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 World Time (Keyless). 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	World Time (Keyless) MCP
Server ID	`019d7625-29d2-7284-8c6e-a7e689ac9250`
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/world-time-keyless.