

M C P S E R V E R

N O C O D E

C L O U D H O S T E D

Local Time Converter MCP

Handle timezone math and timestamp validation without manual errors.

Local Time Converter handles the heavy lifting for temporal math. It lets your AI client convert specific moments between UTC offsets, find the exact hour gaps between regions, and check if your timestamp strings follow ISO 8601 standards. Instead of you manually calculating offsets, your agent does the math instantly.

A+ Quality Score 100/100

timezone

utc

iso8601

time-conversion

temporal



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Local Time Converter MCP

4 tools available

Cloud-hosted on Vinkius

You shouldn't have to waste time doing mental math to figure out if a meeting in Tokyo works for a team in London. This MCP gives your AI client the precision it needs to manage time across different UTC offsets. You can hand off complex temporal tasks, like shifting a specific time from one region to another or determining if one timezone is ahead of or behind another. It also acts as a validator, ensuring your timestamp strings are correctly formatted before you use them in code or logs. Whether you are coordinating global schedules or debugging timestamp data, this tool provides the exact math required to keep your temporal data consistent.

Core Capabilities

01 — Timezone Conversion

Your agent uses this to move a specific time from one offset to another.

02 — Offset Math

The AI calculates the exact hour and minute gap between two regions.

03 — Format Validation

Your client checks timestamp strings against ISO 8601 standards.

04 — Temporal Comparison

The tool identifies if one timezone is ahead of or behind another.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/local-time-converter — connect your AI agent in three steps.

- 01
- Connect the MCP to your client like Claude or Cursor via Vinkius.
- 02
- Ask your agent to perform a time conversion or validation task.
- 03
- The agent calls the specific tool needed for the math.
- 04
- The tool returns the precise time, difference, or validation result.
- 05
- Your agent uses that data to answer your request.

Connecting this MCP to your client gives your agent immediate access to temporal math tools.

Built For

This is for anyone managing data or schedules across multiple timezones. It's built for users who need to hand off precise temporal math to an AI agent.

Software Developers

Hand off timestamp validation and offset calculations to your agent.

Project Managers

Let your AI client calculate meeting windows for global teams.

Data Analysts

Use the tool to verify the integrity of ISO 8601 timestamp strings.

What Changes When You Connect

- 01
- Eliminates manual timezone math errors.
- 02
- Provides instant ISO 8601 format verification.
- 03
- Automates the calculation of UTC offset differences.

04 Enables precise time shifting between global regions.

Real-World Applications

Global Meeting Coordination

Your agent calculates the correct local time for participants in different UTC offsets.

Data Integrity Checks

The AI validates that incoming timestamp data follows the ISO 8601 format.

Log Analysis

Use the tool to compare time gaps between different system logs.

Scheduling Automation

The agent determines if a specific timezone is ahead of or behind your local time.

Patterns to Avoid

Manual Timezone Math

❌ AVOID

You ask your agent to guess if 3 PM in New York is the same as 8 PM in London without checking the offset.

✓ INSTEAD

Use ``convert_time_between_offsets`` to get the exact time shift between specific UTC offsets.

Unchecked Timestamp Strings

❌ AVOID

You feed raw, messy log data into a script and hope the date format is correct.

✓ INSTEAD

Run your strings through ``validate_iso_format`` first to ensure they meet ISO 8601 standards.

Vague Temporal Logic

❌ AVOID

You tell your agent to find out if a team in Tokyo is ahead of a team in Los Angeles.

✓ INSTEAD

Use ``get_timezone_relation`` to get a definitive answer on whether one zone is ahead or behind another.

The Right Fit

Connect this MCP if your workflow involves coordinating global schedules or validating timestamp data. If you only work within a single timezone and never touch ISO 8601 strings, you don't need this. The decision comes down to whether you want to trust your agent with manual temporal math or ensure it uses verified offsets.



Local Time Converter: 4 Precision Tools

These tools handle UTC offset math, ISO 8601 validation, and timezone comparisons.

#	TOOL	DESCRIPTION
01	calculate_offset_difference	This tool finds the exact difference in hours and minutes between two specific UTC offsets.
02	convert_time_between_offsets	Use this to shift a specific date and time from one UTC offset to another.
03	validate_iso_format	This tool checks if a given timestamp string follows the ISO 8601 standard.
04	get_timezone_relation	This tool determines if one timezone is ahead of, behind, or equal to another.

See It in Action

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

U What time will it be in London (+01:00) if it is 14:00 in New York (-05:00)?



It will be 20:00 in London.

U How many hours difference is there between +05:00 and -03:00?



There is an 8 hour difference between these two offsets.

U Is '2023-12-25T12:00:00Z' a valid ISO 8601 timestamp?



Yes, the timestamp is valid.

Frequently Asked Questions

01 How do I use this MCP in Claude?

You connect to the MCP through the Vinkius platform. Once connected, your agent in Claude can automatically call the time conversion tools when you ask about timezones.

02 Can this MCP validate ISO 8601 timestamps?

Yes. It includes a specific tool to check if a timestamp string follows the ISO 8601 format.

03 Does it handle differences in minutes?

Yes. The tool for calculating offset differences provides the exact gap in both hours and minutes.

04 What happens if I need to compare two timezones?

You can use the tool to determine if one timezone is ahead of, behind, or equal to another.

05 Do I need to host this myself?

No. Vinkius hosts and manages the MCP for you, so it is ready to use immediately after you connect.

06 What format should the timestamp be in?

Timestamps must follow the ISO 8601 standard for accurate conversion.

07 How do I specify the UTC offset?

Offsets should be provided with a sign and HH:mm format, such as '+05:00' or '-08:00'.

08 Can I compare two different timezones?

Yes, you can use ``get_timezone_relation`` to determine if one zone is ahead of, behind, or equal to another.

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 <code>"local-time-converter": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Local Time Converter 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 Local Time Converter. 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	September 2026
MCP Server	Local Time Converter MCP
Server ID	01a0d0dd-5aa6-728d-b097-dfe7232564d6
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/en/ai-agent-connect/local-time-converter.