MCP SERVER | NO CODE | CLOUD HOSTED

Date Utils Engine MCP for AI Agents

Calculating Date Differences and Global Timezone Shifts

Date Utils Engine provides deterministic calendar math for AI agents. It stops LLMs from hallucinating dates, missing leap years, or failing timezone conversions. Your agent offloads all complex time calculations to this engine, guaranteeing 100% accurate date arithmetic and reliable global timezone shifts.

A+ Quality Score 100/100

date-math | timezone-conversion | leap-year | calendar-logic | javascript-library | deterministic-calculation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/date-utils-engine

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.

Date Utils Engine MCP

4 tools available
Cloud-hosted on Vinkius

LLMs struggle with consistent timekeeping; they treat calendar math as text tokens, not actual clock data. This MCP brings the industry-standard `date-fns` logic directly into your agentic workflows. You can trust it to handle complex dates, reliably adding or subtracting days, months, and years while maintaining perfect calendar accuracy. Need to shift a date? Simply provide an ISO timestamp and an IANA timezone identifier. The engine automatically accounts for Daylight Saving Time changes, turning messy UTC times into precise local clock readings. When you connect Date Utils Engine through the Vinkius catalog, your AI client gains reliable mathematical certainty for time-sensitive tasks, letting you write code that just works.

Core Capabilities

01 — Add or Subtract Calendar Days

Calculate a precise future or past date by adding or subtracting a specific number of calendar days from any given starting date.

02 — Determine Time Differences Between Dates

Get the exact time delta between two dates, providing counts in days, hours, minutes, and seconds.

03 — Validate Raw Date Strings

Take a raw date string and safely parse it to confirm it represents an actual, valid calendar date object.

04 — Convert Timezones Reliably

Shift any UTC or ISO date to a specific IANA timezone (like America/Los_Angeles), automatically adjusting for Daylight Saving Time rules.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/date-utils-engine** — connect your AI agent in three steps.

01 Start by giving your AI client the initial data: an ISO formatted date string and the mathematical operation required, like 'add 45 days' or 'convert to America/Sao_Paulo'.

02 The MCP sends this request to the deterministic engine, which performs the complex calendar math locally, bypassing typical LLM time errors.

03 You get back a confirmed, accurate date object or delta that accounts for every leap year and timezone rule in play.

The bottom line is your agent doesn't guess about dates; it runs them through reliable, industry-standard calendar logic.

Built For

Any developer or operations engineer who works with global data and time zones. If your workflow involves tracking deadlines across different continents, you need this MCP. Stop wasting time debugging date math errors in production code.

DevOps Engineer

Running validation checks on system logs that contain timestamps from multiple global regions.

Product Manager

Scheduling features or user rollouts based on specific, time-zone aware market launch dates.

Data Scientist

Calculating the exact temporal distance between historical data points collected from varying global sources.

What Changes When You Connect

01 Avoids LLM calendar hallucinations entirely. Instead of guessing dates, your agent uses `add_days` to move a date forward or backward with guaranteed accuracy.

02 Manages complex time zones effortlessly. The `to_timezone` tool converts any UTC timestamp into the correct local time for an IANA zone, even across DST changes.

03 Ensures data integrity upfront. Use `parse_date` to validate messy inputs before running critical calculations like those from `diff_dates`.

04 Saves debugging time. By offloading math, your team stops spending hours fixing date-related bugs that are notoriously difficult to track down.

05 Maintains determinism. The engine executes as native V8 JavaScript, meaning the output is predictable and consistent every single time you call `add_days` or `to_timezone`.

Real-World Applications

Calculating a Project Milestone Date

A PM needs to know when a task starting on March 1st will finish after exactly 60 working days. The agent uses `add_days` to calculate the correct end date, avoiding calendar errors that would push the deadline incorrectly.

Comparing Event Timelines Globally

A data analyst must compare two event timestamps recorded in UTC and one local time zone (e.g., Europe/Berlin). The agent uses `to_timezone` to normalize both dates, providing a single, accurate comparison delta using `diff_dates`.

Validating User Input Forms

A developer needs to check if user-submitted date strings are actually valid before processing them. The agent runs the input through `parse_date`, instantly catching any malformed or non-existent dates, preventing system crashes.

Analyzing Time Spans in Historical Data

A researcher collects two records with timestamps from different time zones. The agent uses `to_timezone` to align both records to a single standard (e.g., UTC) before using `diff_dates` to get the precise span.

Patterns to Avoid

Relying on LLM Date Math

✗ AVOID

Asking an agent, 'What is 30 days after November 1st?' and getting a date that ignores the actual number of days in October or November.

✓ INSTEAD

Use `add_days` to move the calendar forward. This tool handles all month lengths and year boundaries correctly, giving you reliable results.

Ignoring Timezone Shifts

✗ AVOID

Passing a UTC date into an agent and expecting it to represent local time in America/Sao_Paulo without specifying the zone.

✓ INSTEAD

Always run the timestamp through `to_timezone`. This ensures Daylight Saving Time shifts are accounted for, giving you the correct local clock reading.

The Right Fit

Use this MCP if your application's logic relies on perfect temporal accuracy. Specifically, if you need to calculate date intervals or convert times across different global zones, this is your solution. Don't use it just because you handle dates; the math has to be deterministic. If all you need is basic text extraction from a calendar image, skip this MCP and look at an OCR tool instead. However, if you are comparing two dates (`diff_dates`) or needing to shift timezones (`to_timezone`), this engine is non-negotiable.

Date Utils Engine for AI Agents: Solving Calendar Math Problems

Today, date calculations are a huge manual pain point. Every platform needs to handle leap years, month lengths, and timezone offsets manually or via brittle API calls. This leads to bugs where an agent might calculate the wrong deadline because

With Date Utils Engine, your AI client doesn't guess; it calculates. You simply ask for a date change using `add_days` or convert a timestamp using `to_timezone` . The result is mathematically certain, eliminating hours of debugging time.

it forgot November only has 30 days, or it miscalculates a time shift due to DST.

Date Utils Engine for AI Agents: Accurate Timezone Conversion

Previously, setting up global workflows meant writing complex logic to handle every major timezone and the specific rules around DST. This required multiple conditional checks that were prone to failure.

Now, you just pass the date and the IANA zone identifier. The MCP handles all the messy complexity internally. Your agents get reliable time data instantly.

Date Utils Engine: 4 Tools for Accurate Date Calculations

Use these four specialized tools to add/subtract days, calculate time deltas, parse raw dates, and convert timestamps across any global timezone.

#	TOOL	DESCRIPTION
01	`add_days`	Adds or subtracts a specific number of calendar days from a date, guaranteeing 100% accuracy.
02	`diff_dates`	Calculates the exact time difference between two dates and returns the delta in days, hours, minutes, and seconds.
03	`parse_date`	Takes a raw date string and safely parses it, confirming that it is a valid calendar date object.
04	`to_timezone`	Converts an ISO date string to any specific IANA timezone, correctly handling Daylight Saving Time shifts automatically.

See It in Action

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

U What is the exact date 90 calendar days from today's date?

Calculation:
• Start Date: October 12, 2024
• Days Added: 90
Resulting Date: December 10, 2024

(This calculation correctly accounts for the varying length of months and year boundaries.)

U Convert this UTC time to New York time.

Input Timestamp: `2026-07-15T18:00:00Z`
Target Timezone: America/New_York

✅ **Local Time Shift:** 2026-07-15 14:00:00-04:00

(The shift correctly applies the Daylight Saving Time offset for July.)

U How many days are between January 1, 2023 and March 1, 2025?

Date Difference Calculation:
• Start Date: 2023-01-01
• End Date: 2025-03-01
Difference: There are exactly **791** days between the two dates. (This calculation correctly includes leap years.)

Frequently Asked Questions

01 **Does Date Utils Engine help my AI agents with time zone changes?**

Yes, it reliably converts any global timestamp to a specific local timezone using the IANA format. It automatically adjusts for Daylight Saving Time shifts, so you never have to worry about incorrect offsets again.

02 **Can Date Utils Engine calculate dates that span multiple years or months?**

Absolutely. The engine handles complex date arithmetic, letting you add or subtract days across year boundaries and through varying month lengths with guaranteed accuracy.

03 **Why should I use this MCP instead of just telling my agent the math?**

Because LLMs are bad at calendar math. This MCP offloads date logic to a deterministic engine, meaning the results are mathematically certain and reliable in production code.

04 **Is Date Utils Engine useful for calculating time differences between two dates?**

It's perfect for that. You can use it to determine the exact number of days, hours, or minutes separating any two given timestamps, including leap year considerations.

05 **What if I have a messy date string? Can Date Utils Engine fix it?**

The engine includes a tool that safely parses raw date strings. It validates the input first, ensuring you are working with an actual calendar date before any calculations begin.

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 `"date-utils-engine": { "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

Date Utils Engine 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 Date Utils Engine. 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	Date Utils Engine MCP
Server ID	`019e3886-6791-701a-89e2-13445dab8623`
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/date-utils-engine.