MCP SERVER | NO CODE | CLOUD HOSTED

Recurring Event Expander MCP for AI Agents

Calculating Recurring Schedule Dates for Calendar Management

The Recurring Event Expander takes complex calendar math and turns it into simple lists of dates. If you deal with recurring schedules—like 'the third Tuesday of every month'—this MCP calculates all the actual occurrence dates in a given timeframe, validates rules, and translates technical jargon into plain English summaries.

A+ Quality Score 100/100

rrule | rfc5545 | calendar | scheduling | dates | automation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/recurring-event-expander

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.

Recurring Event Expander MCP

3 tools available
Cloud-hosted on Vinkius

Dealing with event recurrence rules (RFC 5545) can feel like decoding ancient calendar text. This MCP handles that complexity for your AI client. You don't have to worry about the intricate logic of intervals, specific weekdays, or exclusions. Instead, you just give it a technical rule string and ask questions. The system calculates every single date an event will occur within a defined window, giving you clean lists ready for use. It also verifies if your rules are even structurally sound, flagging errors before they break your calendar logic. Need to explain the rule to a non-technical manager? Use its summary feature; it translates deep technical patterns into plain language. When integrating this capability with Vinkius, your AI agent gets access to specialized scheduling knowledge, making complex calendar data instantly manageable.

Core Capabilities

01 — Calculate specific occurrence dates

The MCP takes a recurrence rule and calculates every single date the event must happen within a specified start and end window.

02 — Validate calendar rule syntax

It checks an RRULE string to make sure the structure is correct, preventing errors before you try to use the data.

03 — Summarize complex rules into plain English

The MCP translates dense technical recurrence patterns into simple, human-readable descriptions of what the schedule means.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/recurring-event-expander** — connect your AI agent in three steps.

01 First, you pass the recurring event rule (the RRULE string) and the date range to your AI agent.

02 The MCP processes this data, using its tools to either calculate all dates, validate the syntax, or generate a clear summary based on your request.

03 Your agent receives structured output—either an accurate list of specific dates, confirmation that the rule is valid, or a simple English explanation.

The bottom line is, it takes calendar jargon and gives you actionable, error-proof date information.

Built For

This MCP is essential for Operations Analysts, Project Managers, and Data Engineers who spend too much time manually calculating or verifying complex scheduling patterns. If your job involves coordinating events that repeat on specific days (like quarterly board meetings or monthly cycles), this tool saves hours of debugging.

Operations Analyst

They use the MCP to verify if a newly drafted event schedule adheres to historical pattern rules, ensuring no dates are missed.

Project Manager

They rely on this tool to calculate all required milestones and meeting dates for multi-phase projects that repeat over time, like annual compliance audits.

Data Engineer

They use the MCP to validate complex date fields pulled from external sources, ensuring that calendar recurrence rules are syntactically sound before database ingestion.

What Changes When You Connect

01 Stop guessing dates. Use `expand_occurrences` to get a precise, confirmed list of every event date within your required window.

02 Eliminate calendar errors before they happen. Run `validate_rrule` to guarantee that any recurrence string you work with is syntactically correct.

03 No more jargon. The MCP lets your agent run `get_rrule_summary`, instantly translating complex technical rules into language anyone can understand.

04 It handles advanced logic like exclusions (EXDATE) and additions (RDATE), ensuring your calculated dates are comprehensive and accurate.

05 The ability to process varied patterns, including BYDAY and BYMONTHDAY, means it works for nearly every type of repeating schedule you encounter.

Real-World Applications

Determining Quarterly Board Meeting Dates

A user needs to know all the dates for a meeting that happens quarterly on the last Friday. They ask their agent, and it uses `expand_occurrences` to provide a clean list of specific future dates.

Checking Legacy System Data Integrity

A data engineer receives 50 old event records with different recurrence rules. Instead of manually checking them all, they use `validate_rrule` on the entire batch to pinpoint which strings are technically broken.

Explaining a Client's Complex Subscription Cycle

A sales rep needs to explain how a subscription repeats every 3 months and is excluded in August. They run `get_rrule_summary`, and the agent delivers a simple, non-technical explanation for the client.

Patterns to Avoid

Treating RRULE strings as plain text

✗ AVOID

A user pastes an RRULE string into a general chat prompt and asks, 'What are these dates?' The AI simply repeats the jargon without calculating anything.

✓ INSTEAD

Don't just paste it. Use `get_rrule_summary` first to understand what the rule means, then use `expand_occurrences` with specific start/end dates for concrete results.

Manually calculating date ranges

✗ AVOID

A project manager spends an hour trying to map out 18 months of repeating events by hand in a spreadsheet, risking calculation drift.

✓ INSTEAD

Pass the rule directly. Use `expand_occurrences` with your required window dates and let the MCP handle all the complex interval math instantly.

Assuming every calendar pattern is valid

✗ AVOID

A data import fails because a source system used non-standard date formatting, causing the entire job to halt.

✓ INSTEAD

Always run `validate_rrule` first. It catches structural errors in the input string, allowing you to fix just the bad records without failing the whole process.

The Right Fit

Use this MCP if your core problem revolves around interpreting or calculating dates defined by complex recurrence rules (RFC 5545). If you need a list of specific dates within a window, use `expand_occurrences`. If you just need to know *what* the rule means in simple words, run `get_rrule_summary`. Don't use this MCP if your problem is simply finding a date range (use standard calendar tools) or if your data structure isn't already using an RRULE format. You're dealing with the math of repetition, not just single dates.

Using Recurring Event Expander for Scheduling Date Logic

Manually calculating recurring events is a nightmare. You open your calendar and try to map out everything that happens every third Friday, or every month on the 25th unless it's an exclusion date. It’s tedious copy-pasting between tabs, constant cross-referencing of dates, and spending hours just confirming if the rule itself is even valid.

With this MCP, you hand off the logic entirely. You send your agent the technical pattern, and it handles all the math instantly. You get a clean, verifiable list of every single date that matters, freeing you up to focus on scheduling content instead of calendar arithmetic.

Interpreting Event Schedules with Recurring Event Expander

When talking about schedules, people don't care about RFC 5545 standards; they care what it means. You might have to write a dense paragraph explaining that an event occurs 'on the third Tuesday of every month, excluding February.' That’s technical and confusing.

Now you just run `get_rrule_summary`. The agent returns simple English: 'This event happens on the third Tuesday of each month.' It's immediate clarity. You get to communicate complex scheduling ideas simply.

3 Tools for Recurring Event Scheduling Date Calculation

Use these tools to validate, summarize, and expand highly technical recurrence rules into concrete dates your agent can use immediately.

#	TOOL	DESCRIPTION
01	`expand_occurrences`	Calculates and returns a list of specific occurrence dates based on an RRULE string within a given date range.
02	`get_rrule_summary`	Generates a plain English summary that explains the meaning of a complex recurrence rule.
03	`validate_rrule`	Checks and confirms whether an input RRULE string follows proper calendar syntax standards.

See It in Action

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

U I need all dates for a training session that runs every Monday, Wednesday, and Friday from January 20th through February 15th.

Occurrences Found:

- January: 22nd, 23rd, 24th, 29th, 30th
- February: 1st, 2nd, 5th, 6th, 7th, 12th, 13th, 14th

These dates are calculated based on the 'weekly M/W/F' pattern within your requested window. Let me know if you need to adjust the start or end month.

U Is this rule correct: FREQ=MONTHLY;BYDAY=2TH? What does it mean?

Validation Status: Valid

This rule means 'The second Thursday of every single month.' It's a standard, recognized pattern. You can trust it for scheduling.

U Explain this schedule: FREQ=YEARLY;BYMONTH=12;BYMONTHDAY=25, EXDATE=RDATE:2024-12-25

Plain English Summary:

This event happens yearly on December 25th. However, the date in 2024 has been manually removed from the schedule (it's an exclusion).

Next expected occurrence: December 25th of next year.

Frequently Asked Questions

01 **How does Recurring Event Expander handle complex recurring schedules?**

It calculates difficult patterns like 'the third Monday, excluding holidays' into concrete dates. You just provide the rule and a date range, and it outputs every single day that event is scheduled to happen.

02 **Can I use Recurring Event Expander if my schedule pattern isn’t perfect?**

Absolutely. Before calculating anything, you can run a validation check on the rule itself. This tells you instantly if there are any structural errors in your calendar data that need fixing.

03 **What happens if I don’t understand the technical recurrence rules?**

The MCP has a summary tool that translates jargon into plain English. You give it the complex string, and it tells you exactly what the schedule means to a non-technical person.

04 **Is Recurring Event Expander limited to specific types of calendars?**

No. It adheres to the global RFC 5545 standard for recurrence rules, meaning it handles virtually every established calendar pattern used across professional scheduling tools.

05 **Does this MCP work with dates from different time zones?**

The tool focuses on date calculation based on the provided range. While proper timezone handling is key for final display, the core recurrence logic works reliably once you define your operational window.

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 `"recurring-event-expander": { "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

Recurring Event Expander 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 Recurring Event Expander. 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	July 2026
MCP Server	Recurring Event Expander MCP
Server ID	`019f266a-4455-7304-923d-40ec8a73979a`
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/recurring-event-expander.