MCP SERVER | NO CODE | CLOUD HOSTED

Postmark MCP

Analyze delivery performance and track bounces instantly.

Postmark MCP lets your AI agent manage and analyze every part of your transactional email pipeline. Send emails using pre-set templates or raw HTML; check out detailed bounce logs (Hard/Soft Bounces); review server analytics like open rates, and query message history—all without logging into the Postmark UI.

A+ Quality Score 98.33/100

transactional-email | smtp | email-delivery | bounce-tracking | api-integration | email-templates

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/postmark-alternative

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.

Postmark MCP

9 tools available
Cloud-hosted on Vinkius

Your AI agent uses this MCP to fully manage your application's transactional email flow. Instead of relying on a web dashboard or writing boilerplate code, you ask your client to perform actions directly against your mail service. For instance, if you need to know why some users didn't get an update, you can query the bounce logs and pinpoint whether it was a hard failure or a block. You can also pull templates by name or ID, letting your agent review the raw HTML to make sure dynamic variables like `first_name` are placed correctly. The entire system is accessible through Vinkius, the #1 MCP Catalog, meaning you connect once from any compatible client and get full control over your email notifications right where you're already working.

Core Capabilities

01 — Send emails via templates or raw HTML

Your agent sends transactional messages using either a pre-built Postmark template or custom, plain HTML/text.

02 — Analyze bounce failure logs

The MCP retrieves detailed data on why specific email recipients failed delivery, distinguishing between hard bounces and soft bounces.

03 — Review template structure

You can fetch the source code of any Postmark template to check its required variables or visual layout.

04 — Check delivery analytics overview

The agent pulls general stats on outbound messages, including open rates and overall system health metrics.

05 — Search specific message history

You query the full send history by filtering criteria like sender email address or delivery status.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/postmark-alternative** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Postmark Server Token.

02 Your AI client accesses the tools through natural conversation (e.g., 'Check our bounce rate for last month').

03 The agent executes the necessary calls, retrieves the structured data (like a list of failed emails or template code), and presents it directly in the chat.

The bottom line is you get full control over your application's email notifications from anywhere, without needing to open another browser tab.

Built For

Engineers who spend too much time copying data between Postmark's dashboard and their local debugging console. Product managers who need quick proof of concept analytics for a feature launch. System admins tired of manually querying logs to find out why an email failed.

Software Engineer

They use the MCP to trigger test template dispatches from their terminal and review server logs immediately to debug application email flows.

Product Manager

They check bounce rates and verify the visual appearance of critical system emails without having to navigate the Postmark UI.

System Administrator

They query delivery analytics quickly or trace specific email endpoints to get a fast operational overview during an outage.

What Changes When You Connect

01 Stop checking dashboards. Instead, ask your agent to run `search_bounces` and tell you exactly which recipients are bouncing and why. You get the specific error code right in your chat window.

02 When deploying a new feature, use the MCP to call `get_template` on critical emails. This lets you review the raw HTML structure, verifying that all dynamic variables will align perfectly with your codebase before deployment.

03 Need to send a test message? Just ask it to `send_email_with_template`. You can confirm delivery instantly and get back the Message ID, proving the email was dispatched correctly without manual steps.

04 Debugging is faster when you can query history. Use `search_outbound_messages` to filter by status or sender address, giving you a precise view of message flow that's hard to find in an admin UI.

05 Get immediate health checks. The MCP allows querying the `get_outbound_overview` tool so your agent can summarize key metrics like open rates and general delivery performance instantly.

Real-World Applications

A user notices a sudden spike in failed updates.

Instead of diving into the web console, they ask their agent to run `search_bounces`. The MCP immediately filters the logs and shows that 80% of the failures are 'recipient blocked' (Hard Bounces), letting them know the problem is address list hygiene, not code.

A product manager needs to verify a new feature email layout.

They ask for `get_template` on the 'Welcome' message. The agent returns the raw HTML body and lists required variables (e.g., `first_name`), allowing them to approve the design without manual visual checks.

A developer needs to confirm a critical test email sends correctly.

The developer prompts the agent to `send_email` to their QA colleague. The response confirms success and provides a Message ID, letting them commit code knowing the delivery mechanism worked perfectly.

An admin needs an immediate health check on mailing volumes.

The admin asks for the `get_outbound_overview`. The agent pulls the current stats in seconds—a summary of open rates and message volume metrics—saving them 10 minutes of dashboard clicking.

Patterns to Avoid

Confusing general logging with specific failure analysis

✗ AVOID

The developer just looks at the main Postmark 'Activity' log, which shows a message failed but doesn't specify if it was due to a mailbox full error or a permanent block.

✓ INSTEAD

Use `search_bounces` instead. This tool lets you filter by bounce type and gives you the exact reason code for hard bounces (like blocked) versus temporary failures, which is crucial for debugging.

Relying on manual UI interactions to send test emails

✗ AVOID

The developer has to manually switch environments in the Postmark dashboard just to fire off a quick test email to themselves.

✓ INSTEAD

Use `send_email` or `send_email_with_template`. This lets your agent dispatch the mail instantly using parameters you define, keeping all debugging confined to your AI chat window.

Assuming template variables are correct

✗ AVOID

The product team deploys a new email and finds that the 'support link' variable is missing or formatted incorrectly in production.

✓ INSTEAD

Before deployment, run `get_template` on the affected message. This tool exposes the raw HTML source code, letting you inspect every required dynamic variable (`action_url`, etc.) for accuracy.

The Right Fit

Use this MCP if your primary need is deep operational visibility into transactional email failures and performance metrics. Specifically, use it when you need to know *why* an email failed (bounces), or when you must programmatically verify template structure before sending. If your goal is simply to send a one-off notification without

needing any analytics or historical tracking, then a basic SMTP credential might suffice. But if you are building robust applications that rely on reliable delivery and need to debug failures by type, this MCP's ability to `search_bounces` and `search_outbound_messages` is unmatched in an agent environment. Don't use it just because you want a dashboard; use it because the data structure (the bounce reason codes) is necessary for your logic.

The Headache of Tracking Failed Emails

Today, when an email fails to deliver, you're usually forced into a painful cycle. You have to switch tabs, navigate through the 'Bounces' section in the web UI, manually filter by date range, and then visually scan hundreds of entries just to find out if the failure was temporary or permanent. It’s slow, it’s tedious, and you risk missing critical error codes.

With this MCP, that process vanishes. You simply tell your agent what you're looking for—for example, 'Show me all hard bounces from last week.' The tool runs `search_bounces`, pulls the data into the chat context, and gives you a clean, actionable list of failures without any manual clicking or dashboard switching.

Get Template Details with get_template

Before deploying an email that uses custom variables, the old way required logging into the Postmark console and finding the template editor. You'd have to click around until you could view the raw HTML source code just to verify if a variable like `first_name` was actually used correctly in the body.

Now, all you do is ask your agent to run `get_template`. It returns the full source code directly to your chat. You can instantly confirm that the template expects and uses every variable your application sends, making deployments faster and safer.

Postmark Alternative MCP with 9 Tools

These tools let you query server details, manage templates, send messages, and perform deep analysis on bounce logs and message history using natural language commands.

#	TOOL	DESCRIPTION
01	`get_bounces_overview`	Retrieves a summary of your overall bounce statistics.
02	`get_server`	Pulls basic operational details about the connected Postmark server instance.
03	`get_template`	Fetches the specific code and structure for a single named email template.
04	`list_templates`	Provides an index of all available templates configured on your server.
05	`get_outbound_overview`	Retrieves general metrics and performance data for all outgoing messages.
06	`search_bounces`	Searches your bounce logs, allowing filtering by type (Hard/Soft), specific email address, or message ID.
07	`search_outbound_messages`	Queries the historical record of sent emails by filtering based on recipient, sender, or delivery status.
08	`send_email`	Sends a basic transactional email using specified From/To addresses and either plain text or HTML content.
09	`send_email_with_template`	Dispatches an email by applying the logic of a specific, pre-configured Postmark template.

See It in Action

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

U Check the bouncing metrics for our transactional emails.

I've analyzed your bounce metrics. The server currently has 14 Hard Bounces and 3 Soft Bounces within the query scope. Overall health looks good! Want me to list the specific emails?

U Can you fetch the HTML body layout of our 'Welcome' template?

I found your 'Welcome' template (ID: 809214). It expects three variables: `first_name`, `action_url`, and `support_email`. The HTML renders a classic logo-centric email layout.

U Send a standard test email to alice@example.com using our current Postmark setup.

The email has been processed! Postmark returned success mapping with a Message ID `f8a847a9-2169-45e0-...` indicating instantaneous dispatch.

Frequently Asked Questions

01 **How do I check bounce metrics using Postmark MCP?**

You use `get_bounces_overview` for a general summary of current bounces. If you need to investigate specific failures, run the `search_bounces` tool and filter by type (Hard or Soft) to pinpoint the exact error.

02 **Can Postmark MCP send emails with my custom code?**

Yes. You can use the `send_email` tool, which accepts raw HTML or plain text bodies, allowing you to bypass templates for specific testing scenarios.

03 **What if I want to check all my sent messages in Postmark MCP?**

Run the `search_outbound_messages` tool. This lets you filter by recipient or status, giving you a comprehensive look at your message history without manually browsing folders.

04 **Does Postmark MCP help with template debugging?**

Absolutely. Use `get_template` to fetch the raw HTML source code for any template. This lets you visually and structurally verify that all required variables are correctly implemented in the body.

05 **Is postmark-alternative MCP faster than using the Postmark UI?**

Yes, because it eliminates context switching. You stay within your preferred AI client environment (Claude, Cursor, etc.) and execute commands instantly via conversation rather than navigating a separate web interface.

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 `"postmark-alternative": { "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

Postmark 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 Postmark. 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	Postmark MCP
Server ID	`019d8470-ad96-72d1-9e8e-23ceab745a14`
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/postmark-alternative.