MCP SERVER | NO CODE | CLOUD HOSTED

MailSlurp MCP

Automate Email Testing with AI Agents

MailSlurp lets your AI agent manage temporary inboxes and automate email testing. Instead of manually signing up for disposable addresses, you generate random or custom emails on the fly. Your agent can monitor these virtual mailboxes, wait for specific confirmation emails to arrive, retrieve full message bodies, and even send follow-up messages programmatically.

A+ Quality Score 100/100

email-testing | virtual-inbox | disposable-email | automated-testing | email-automation | message-parsing

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/mailslurp

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

MailSlurp MCP

8 tools available
Cloud-hosted on Vinkius

Testing web apps means dealing with email flow—user signups, password resets, welcome alerts. Manually checking these flows is a huge pain point. This MCP connects your agent directly to MailSlurp's powerful infrastructure. It lets you build complex test suites entirely through natural conversation. Your AI client can create temporary inboxes and keep them ready for testing. Need to see if an email arrived 30 minutes from now? Your agent can long-poll the inbox until it shows up, giving you immediate results without writing polling code. You'll get access to detailed message headers, allowing you to verify not just that the email was sent, but how it traveled through the system. Because this MCP is hosted on Vinkius, you connect once from any compatible client and immediately gain access to this whole suite of inbox operations.

Core Capabilities

01 — Generate disposable inboxes

Your agent creates new random or custom email addresses for temporary use.

02 — Monitor incoming emails

The system waits for the next email to arrive, or monitors until a specific count of messages has been received.

03 — Inspect message content

You retrieve and view all details for any message, including both plain text and full HTML bodies.

04 — Send outbound emails

The agent sends new emails directly from one of your temporary addresses to a recipient.

05 — Manage inboxes

You list, inspect, or permanently delete entire virtual mailboxes when testing is done.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the MailSlurp MCP and enter your unique API key.

02 Next, tell your agent what you need—for example, 'Create a new random inbox' or 'Wait for an email with the subject verify'.

03 Finally, your AI client executes the necessary actions, giving you real-time status updates on inbox creation, message arrival, and content retrieval.

The bottom line is that your agent treats these temporary mailboxes like a live API endpoint, letting you manage complex email logic without writing any boilerplate code.

Built For

QA Engineers and Developers who spend too much time manually signing up for test accounts or checking development logs. Product teams monitoring critical communication paths also benefit from this.

QA Engineer

Running end-to-end tests on user onboarding flows, verifying that transactional alerts like password resets arrive correctly and contain the right content.

Developer

Quickly inspecting test emails sent from microservices directly inside their IDE to debug formatting or missing metadata.

Product Manager

Auditing automated communication paths after a feature launch, ensuring that all necessary alerts and confirmations are delivered as expected.

What Changes When You Connect

01 Stop manually checking email logs. With this MCP, your agent handles the asynchronous waiting period using 'wait_for_latest_email' or 'wait_for_email_count', giving you instant results right in your natural language flow.

02 Never worry about test accounts again. Use 'create_random_inbox' to get a disposable address instantly for every single scenario, keeping your testing clean and repeatable.

03 Get full context on delivery failures. The 'get_email_details' tool doesn't just give you the body; it pulls detailed technical headers so you can verify exactly how the email traveled.

04 Test both directions in one session. You can monitor incoming messages using 'wait_for_latest_email', and then use 'send_email_from_inbox' to send a reply, all without switching tools or interfaces.

05 Clean up after testing is done. Use 'delete_specific_inbox' and 'list_all_inboxes' to manage your infrastructure efficiently, preventing mailbox clutter.

Real-World Applications

Verifying a user signup flow

A QA engineer asks their agent to monitor the inbox. The agent uses 'wait_for_latest_email' and then 'get_email_details' to confirm that both the welcome email and the verification link were received, ensuring the full transactional path works.

Simulating a batch notification

A product team needs to test if their system sends five follow-up alerts. They ask the agent to 'wait_for_email_count' until the inbox hits 5, confirming the automated communication path is correct.

Debugging a complex integration

A developer needs to test if Service A and Service B both send confirmation emails. They use 'create_random_inbox' and then monitor it with the agent, checking for multiple arrivals using 'list_inbox_emails'.

Testing automated replies

A user wants to simulate a follow-up action after receiving an alert. They use the agent to read the content via 'get_email_details', and then execute a reply using 'send_email_from_inbox'.

Patterns to Avoid

Writing complex email logic in code

✗ AVOID

Writing dozens of lines of Python or JavaScript to handle polling, retries, and parsing email bodies just to check a simple sign-up flow.

✓ INSTEAD

Use the MailSlurp MCP. Your agent handles the complexity: use 'wait_for_latest_email' for real-time checks and 'get_email_details' to parse the content, keeping your code clean.

Forgetting cleanup steps

✗ AVOID

Running a test suite that leaves dozens of temporary mailboxes running, cluttering your account and making it hard to track which addresses are active.

✓ INSTEAD

After testing, use 'list_all_inboxes' to see what you created, then run 'delete_specific_inbox' on everything finished. Keep things tidy.

Assuming email content is plain text

✗ AVOID

Writing a script that only looks for keywords in the visible body of an email and misses important metadata like the sender or unique message ID.

✓ INSTEAD

Use 'get_email_details' to pull all available technical headers. This ensures you verify both the content *and* the delivery parameters.

The Right Fit

You should use this MCP if your primary pain point is dealing with asynchronous, time-delayed communications in your testing environment—specifically emails. If you need to simulate a user signing up and waiting for confirmation alerts, or test complex multi-step communication paths (like signup -> welcome email -> payment alert), this is the tool. Don't use it if you simply need to store structured data like usernames or transaction IDs; those belong in a database MCP. Also, don't rely on it just for simple

message sending outside of a testing flow—for general outbound messaging, there are simpler communication tools available. But for controlled, verifiable email *testing*, this is unmatched because the agent can wait and react to real-time mail arrival.

The Problem with Manual Email Testing

Today, if a developer needs to verify a new user registration flow, they typically have to manually sign up for an account. They wait 5 minutes, then copy the temporary email address into their local mail client. After the system sends the confirmation link, they open that message, check the headers, and copy the content—all while juggling multiple browser tabs.

With this MCP, you tell your agent to monitor a new inbox using 'create_random_inbox'. You let the AI client handle the waiting period; it waits for the email to arrive automatically. When it pops up, you get all the data—the body, the headers, and the source—instantly passed back to your natural language prompt.

MailSlurp MCP: Full Inbox Operations

You no longer need separate scripts for monitoring or sending. Your agent can perform a full cycle: it generates the address with 'create_random_inbox', waits patiently until an email arrives using 'wait_for_latest_email', extracts the necessary data via 'get_email_details', and then sends a reply using 'send_email_from_inbox'.

The entire complex, multi-step process of mailbox management is collapsed into one conversation. You just tell your agent what to do.

MailSlurp with 8 Tools

Use these tools together to create sophisticated automated workflows for managing disposable mailboxes and testing communication paths.

#	TOOL	DESCRIPTION
01	`create_random_inbox`	Generates a completely new, random email address for you to use in testing.
02	`delete_specific_inbox`	Permanently removes an entire virtual inbox and all the emails associated with it.
03	`list_inbox_emails`	Retrieves a list of every email that has arrived in a specified mailbox.
04	`get_email_details`	Fetches the complete, detailed content for any single received email, including attachments and headers.
05	`list_all_inboxes`	Displays a list of every virtual inbox you currently manage in your account.
06	`send_email_from_inbox`	Allows the agent to send an outgoing email using one of your temporary addresses and specified content.
07	`wait_for_email_count`	Pauses execution until a specific inbox has reached a predefined total number of messages.
08	`wait_for_latest_email`	Monitors an inbox and returns the most recent incoming email immediately if one is available, or waits for the next one.

See It in Action

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

U Create a new random inbox in MailSlurp.

I've created a new inbox. The address is 'abc-123@mailslurp.com' and the ID is 'inbox-xyz'. You can now use it to receive emails.

U Wait for the next email to arrive in inbox 'inbox-xyz'.

I'm monitoring the inbox. [Connection held] ... I've received an email from 'no-reply@service.com' with the subject 'Verify your account'. Would you like the text body?

U Send an email to 'user@example.com' from inbox 'inbox-xyz'.

I've sent the email to user@example.com from your virtual inbox inbox-xyz. The message has been successfully delivered to the outbound queue.

Frequently Asked Questions

01 **How many inboxes can I manage with the MailSlurp MCP?**

You can list all managed inboxes using 'list_all_inboxes' and create new ones as needed. The system handles your entire mailbox infrastructure.

02 **Can I use MailSlurp to wait for a specific keyword in an email?**

Yes, you can set up complex waiting conditions using 'wait_for_email_count' or by monitoring the inbox until a message arrives that contains your target keywords.

03 **Is the content retrieved by get_email_details plain text only?**

No, 'get_email_details' provides complete information, including both the readable text body and the full source HTML for deep inspection.

04 **What happens to my temporary inboxes after I finish testing?**

You should use the 'delete_specific_inbox' tool. This permanently removes the inbox and all associated emails, keeping your account clean.

05 **Does MailSlurp only work for receiving emails?**

Not at all. You can send test responses or follow-ups using 'send_email_from_inbox', simulating a complete round trip within your tests.

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 `"mailslurp": { "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

MailSlurp 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 MailSlurp. 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	MailSlurp MCP
Server ID	`019d75cd-4f9b-728a-9d29-03528013a166`
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/mailslurp.