MCP SERVER | NO CODE | CLOUD HOSTED

Linear MCP for AI Agents

Manage your software engineering sprints and issue tracking through natural conversation.

Linear MCP lets your AI agent manage your project boards, sprints, and issue tracking without you having to switch tabs. It handles everything from creating tickets and assigning priorities to checking cycle progress and querying team data.

A+ Quality Score 100/100

issue-tracking | sprint-planning | workflow-automation | task-management | team-collaboration | software-development

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/linear

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.

Linear MCP

12 tools available
Cloud-hosted on Vinkius

Imagine you're in the middle of a deep coding session and realize you need to update a ticket's priority or assign a bug to someone else. Usually, that means breaking your flow to open a browser, find the right project, and click through several menus. This MCP changes that by letting your AI agent handle the heavy lifting. Instead of manual data entry, you just tell your agent what needs to happen. It can pull up a list of open issues for your team, tell you how much of the current sprint is left, or even draft a comment on a bug report while you stay focused on your IDE. It turns your AI into a sort of engineering manager that keeps the project moving in the background. You can find this and thousands of other connections in the Vinkius catalog to get your dev stack synced up quickly. It's about making the project management part of your job feel less like a chore and more like a conversation. Think about the time spent hunting for a specific ticket or trying to remember which team owns a particular bug. This MCP removes that friction. You can ask your agent to find every issue related to a specific feature or summarize the current status of a project. It handles the tedious parts of project management so you can spend more time actually shipping code. Whether you're updating a status, adding a mention to a teammate, or checking the completion percentage of a sprint, the AI does the clicking for you. It's about moving from a world of constant tab-switching to a world where your project management is just another part of your natural conversation with your agent.

Core Capabilities

01 — Create new tickets

Generate new issues with specific priorities, labels, and descriptions instantly.

02 — Query existing issues

Search through your backlog using keywords or filter by specific teams.

03 — Update ticket details

Modify assignees, priorities, and statuses of existing tickets via chat.

04 — Check sprint progress

Retrieve the completion percentage and dates for any team's active cycle.

05 — Add comments

Post updates to issues using Markdown and @mentions without leaving your workspace.

06 — Browse projects

View high-level project details and team structures across your entire workspace.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP in the Vinkius catalog.

02 Enter your Linear Personal API Key in the configuration settings.

03 Start asking your agent to manage your tickets or check sprint status.

The bottom line is you get to manage your entire engineering workflow through a chat interface instead of a cluttered dashboard.

Built For

This is for the engineering team who is tired of the manual overhead of project management. It's for the people who want to keep their focus on code while keeping the roadmap updated.

Engineering Manager

Checks sprint health and pulls issue summaries during standups without opening a browser.

Software Developer

Updates ticket statuses and adds comments to bug reports directly from their IDE.

Product Manager

Searches for feature blockers and monitors project milestones via natural conversation.

What Changes When You Connect

01 Stop tab switching: Use `list_issues` and `search_issues` to see what's on your plate without leaving your editor.

02 Faster ticket creation: Use `create_issue` to log bugs or tasks instantly as they come to mind.

03 Better team context: Use `create_comment` to keep discussions updated with mentions while you stay in your flow.

04 Real-time sprint tracking: Use `list_cycles` to see exactly how much of the current sprint is finished.

05 Organized project oversight: Use `list_projects` and `get_project` to get a bird's eye view of your roadmap.

06 Easier team discovery: Use `list_teams` and `list_labels` to quickly find the right categories for your work.

Real-World Applications

Log a bug while coding

A developer wants to log a bug while coding. They ask the agent to create an issue with high priority for the backend team.

Check sprint status

A manager needs a status update. They ask the agent to list the current cycle progress for the mobile team.

Search for blockers

A product lead is looking for blockers. They ask the agent to search for all unresolved issues in the engineering team.

Update ticket details

A team member needs to update a ticket. They ask the agent to change the assignee and status of a specific issue.

Patterns to Avoid

Trying to update issues without IDs

✗ AVOID

Asking the agent to "Change the status of the login bug" without providing a specific identifier.

✓ INSTEAD

Use `get_issue` first to find the unique UUID for the ticket, then use `update_issue` to change the status.

Creating issues in the wrong team

✗ AVOID

Asking to "Create a ticket for the UI team" when the agent doesn't know the specific team ID.

✓ INSTEAD

Use `list_teams` first to get the correct team ID, then provide that ID when calling `create_issue`.

Hardcoding priorities

✗ AVOID

Asking the agent to "Make this urgent" which might not map correctly to the project's priority scale.

✓ INSTEAD

Use `create_issue` with the specific priority integer, such as 1 for Urgent, to ensure it shows up correctly.

The Right Fit

Use this if you need to automate your project management tasks, update tickets via voice or text, and keep your engineering team synced without manual data entry. Don't use it if you need a visual drag-and-drop board; this is for interacting with the data, not replacing the UI. If you only need to read data without any actions, a simple read-only connector might suffice, but this MCP is built for full interaction.

Linear MCP for Agile Issue Tracking and Sprint Management

Every time a developer hits a snag, they have to stop, open a browser, navigate to the project board, find the right ticket, and type out a description. It breaks focus and creates a mental tax that slows down actual coding.

With this MCP, you just tell your agent to log the bug. It handles the metadata, sets the priority, and assigns the team in one go. You get to keep your momentum while the project stays organized.

Speed Up Engineering Workflows with Linear MCP Project Oversight

Manually checking cycle progress or searching for specific labels across multiple teams is a chore that eats up time during standups and planning sessions.

This MCP pulls that data into your chat instantly. You get clear answers on sprint health and project status without the clicks.

Linear MCP with 12 Tools for Agile Project Management

Manage your tickets, cycles, and projects through your AI client using these 12 specific Linear tools.

#	TOOL	DESCRIPTION
01	search_issues	Find specific tickets using keywords or filters for a specific team. This helps you quickly locate tasks without digging through boards.
02	create_comment	Post a new comment on an issue using Markdown and @mentions. This lets you share updates with teammates directly from your chat.
03	create_issue	Generate a new ticket with a title, description, priority, and labels. This makes it easy to log bugs the moment you find them.
04	list_cycles	View all sprint cycles for a team, including their progress and dates. This gives you a quick look at how much work is left in a sprint.
05	get_issue	Fetch every detail for a specific ticket using its ID or human-readable key. Use this to get the full context of a bug or task.
06	get_project	Retrieve the full details and status for a specific project. This helps you see the high-level goals of your work.
07	list_issues	Get a list of all tickets, with an optional filter for a specific team. This is perfect for getting a quick overview of the current backlog.
08	list_labels	See all the labels used for categorizing issues across your workspace. This helps you keep your organization consistent.
09	list_projects	Browse all active projects across multiple teams with a result limit. This lets you see the broader scope of the company's work.
10	list_teams	Discover all teams in your workspace to find the right IDs for queries. Use this to understand the team structure of your organization.
11	update_issue	Modify specific fields on an existing ticket using its unique UUID. This allows you to change status or assignees in seconds.
12	get_viewer	Check which account the current API token belongs to for verification. This is a quick way to confirm your connection is correct.

See It in Action

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

U What's the status of the mobile team's current sprint?

The Mobile team's active cycle is **#47** (Mar 31 — Apr 13) with **62%** completion.

- **Total Issues:** 14
- **Completed:** 5
- **Remaining:** 9

Would you like me to list the remaining open issues for this cycle?

U Create a high priority bug for the backend team called 'API Timeout'.

Done! I've created issue **BE-217** titled 'API Timeout' with **High** priority in the Backend team.

You can view it here: linear.app/yourco/issue/BE-217

U Find all open issues for the engineering team.

I found 7 open issues for the Engineering team:

ID	TITLE	PRIORITY	ASSIGNEE
ENG-142	Fix auth token refresh bug	Urgent	@sarah_dev
ENG-138	Implement pagination	Normal	@mike_dev
ENG-145	Update README	Low	@dev_team
...	...	...	

...

Frequently Asked Questions

01 **Can I use the Linear MCP to create tickets?**

Yes, you can create new issues with a title, description, priority level, and specific labels directly through your AI client.

02 **Does Linear MCP support sprint cycles?**

Yes, it can list all cycles for your teams, including start and end dates as well as the current completion percentage.

03 **Can my AI agent add comments to Linear?**

Yes, your agent can post comments on issues using Markdown and @mentions to keep your team updated.

04 **How do I find my team ID for the Linear MCP?**

You can use the list_teams tool to see all teams in your workspace, which will provide the unique IDs needed for specific queries.

05 **Can I update existing issues using this MCP?**

Yes, you can update any field on an existing ticket, such as the status, assignee, or priority, using its unique identifier.

06 **Is there a way to search for issues with specific keywords?**

Yes, the search_issues tool allows you to find tickets by typing in keywords or filtering by a specific team.

07 **How do I find my Linear Personal API Key?**

Go to [**Linear Settings > API**](https://linear.app/settings/api) and click **Create API Key**. Give it a descriptive name like 'Vinkius MCP' and copy the generated key immediately — it won't be shown again.

08 **Can I create new issues with assignees and labels?**

Yes! Use the `create_issue` tool with the required `team_id` and `title` parameters. Optionally provide `assignee_id`, `priority` (0-4), `description` in Markdown and `label_ids` as a comma-separated list. The agent will return the created issue's identifier and URL.

09 **Does the agent have access to delete issues or modify team settings?**

No. This server focuses on read operations and safe mutations — creating issues, updating fields and adding comments. Destructive actions like deletion or team configuration changes are not exposed, keeping your workspace secure.

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 `"linear": { "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

Linear 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 Linear. 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	Linear MCP
Server ID	`019d8453-d421-73dd-9e4f-c21b18b88228`
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/linear.