MCP SERVER | NO CODE | CLOUD HOSTED

TAPD MCP

Automate your entire agile product workflow.

TAPD brings professional agile development management into your AI client. This MCP lets you automate complex product workflows, handling everything from listing workspaces and creating new stories to tracking bugs and managing daily tasks. Stop clicking through multiple tabs; just talk to your agent about the status of any feature or defect.

A+ Quality Score 100/100

agile-development | bug-tracking | sprint-planning | product-management | issue-tracking

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

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.

TAPD MCP

10 tools available
Cloud-hosted on Vinkius

Managing a software project requires constant context switching—jumping between bug reports, feature backlogs, and task boards. This MCP lets your AI agent take over that whole process. Instead of navigating the complex TAPD enterprise interface, you simply ask questions in natural language. Your agent handles listing all accessible workspaces, finding specific stories for requirements, or even creating a new defect report. It keeps track of which team members are assigned to what tasks and monitors sprint progress without you needing to know the platform's internal structure. By connecting this MCP through Vinkius, your AI client treats TAPD like a conversational product assistant, keeping development pipelines organized whether you use Scrum or Kanban.

Core Capabilities

01 — Discovering Projects and Teams

List all available workspaces and see which team members are assigned to them.

02 — Managing Feature Requirements

Create new user stories or view existing ones, complete with titles and detailed descriptions.

03 — Tracking Quality Defects

List current bugs in a project or create an entirely new bug report to keep quality assurance moving.

04 — Handling Daily Tasks

Create, list, and manage specific granular tasks needed for daily development work.

05 — Monitoring Development Cycles

Review project milestones by listing iterations (sprints) to check delivery schedules.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your TAPD API User credentials.

02 Your AI client connects using the provided keys, granting access to all workspaces.

03 You tell your agent exactly what you need—for example, 'List all bugs for Project X'—and it runs the command.

The bottom line is that your development data moves from a complex web interface into a simple conversation with your AI client.

Built For

This MCP is essential for anyone involved in the product development lifecycle, especially Product Managers and QA Engineers who are tired of manually gathering status updates across multiple dashboards. If you spend more time compiling reports than actually building features, this is for you.

Product Manager

Uses the MCP to audit the product backlog by listing stories and monitoring iteration progress via natural language queries.

QA Engineer

Relies on this MCP to report defects using create_bug and track their resolution status across multiple projects.

Software Developer

Manages their day-to-day work by listing tasks, updating bug statuses, and checking assigned stories directly through the AI agent.

What Changes When You Connect

01 Stop manually listing stories. You can use list_stories and create_story to manage the entire feature lifecycle conversationally, keeping requirements tracked automatically.

02 Defect tracking gets simple. Instead of clicking into bug boards, you just ask the agent to list_bugs or create a new defect using create_bug.

03 Get context instantly. Use list_workspaces to see every project you manage and then get_workspace details without logging into multiple portals.

04 Never lose track of who's doing what. List_members gives instant team visibility, while list_tasks shows exactly which daily activities are pending for the team.

05 Plan sprints effortlessly. You can use list_iterations to check project milestones and ensure your development cycles stay on schedule.

Real-World Applications

The Weekly Status Report

A Product Manager needs a status report combining all features, bugs, and tasks across three different products. Instead of logging into TAPD three times, they ask their agent to list_workspaces first, then run list_stories, list_bugs, and list_tasks for each one. The agent compiles the full, accurate summary in seconds.

The Hotfix Incident

A QA Engineer finds a critical bug right before launch. They don't want to wait until the daily standup. They immediately tell their agent, 'Create a bug in Mobile App V2.' The agent uses create_bug to log it instantly and assigns the correct priority.

Starting a New Feature

A team needs a new feature defined. Instead of drafting an email and manually updating the backlog, the Product Manager just tells their agent, 'Create a story for user authentication.' The agent executes create_story and logs it properly.

Onboarding New Engineers

A new developer needs to see what work is pending. They ask their agent to list_tasks and then list_members to see who they need to connect with, getting a clear picture of the team structure immediately.

Patterns to Avoid

Assuming AI knows your project scope

✗ AVOID

The user asks, 'What bugs are there?' without specifying the workspace or project name. The agent fails because it doesn't know which data set to pull from.

✓ INSTEAD

Always specify location details. First, list_workspaces to narrow down the project, then use list_bugs while referencing that specific workspace context.

Trying to manage everything in one command

✗ AVOID

The user tries to write a single prompt like: 'Update stories and bugs and tasks for Q2.' This is too vague and the agent can't execute anything.

✓ INSTEAD

Break it down. Use list_iterations to confirm the sprint, then run specific commands like list_stories or create_bug separately for clarity.

Confusing tasks with stories

✗ AVOID

The user asks the agent to 'list story details' when they actually need a simple daily checklist of work items.

✓ INSTEAD

Use list_tasks. Stories are high-level requirements (the why); tasks are the small, immediate actions needed to complete the feature (the how).

The Right Fit

Use this MCP if your job requires managing complex, structured data across multiple agile domains—specifically issue tracking, backlog management, and team resource allocation. You need an agent that knows the difference between a 'story' (a high-level requirement) and a 'task' (the daily work item). Don't use it if you just need to read simple documents or manage calendars; those are better handled by dedicated calendar MCPs. If your main pain point is simply sending messages, an email-based tool will suffice. But if your entire workflow lives inside the TAPD platform and requires reading, creating, updating, and listing specific artifacts like bugs or stories, this is mandatory.

The sheer volume of status updates kills productivity.

Every week, your team spends hours manually jumping between the TAPD interface. You open the backlogs to review requirements, switch tabs to check bug statuses, and then jump again to update small tasks. This process isn't just slow; it guarantees you miss critical context because of all the clicking.

With this MCP connected via Vinkius, your agent acts as a single window into that massive platform. You tell it, 'What is the status of authentication?' and it instantly aggregates data from stories, tasks, and bugs without ever making you click anything.

TAPD MCP: Get instant control over features and defects.

The most tedious parts are the repeated searches. You waste time manually listing stories to check requirements, then running list_bugs just to see if a recent defect was missed, all while trying to reconcile who owns which task.

Now you simply ask your agent. It runs list_stories or list_bugs and delivers a clean, actionable summary right in your chat window. You get the data, not the dashboard.

TAPD MCP: 10 Issue Tracking Tools

Use these tools to handle every aspect of the agile lifecycle, from listing all available projects to creating specific bug reports and tracking tasks.

#	TOOL	DESCRIPTION
01	`create_bug`	Generates a brand new defect report within your TAPD workspace.
02	`create_story`	Adds a new feature requirement (story) to the backlog with a title and description.
03	`create_task`	Creates a small, actionable item needed for development work within the workspace.
04	`get_workspace`	Retrieves specific details about a single, selected project workspace.
05	`list_bugs`	Shows all existing defects or bugs recorded in the current development space.
06	`list_iterations`	Retrieves a list of scheduled project sprints and milestones for tracking progress.
07	`list_members`	Lists all team members associated with the workspace to manage assignments.
08	`list_stories`	Displays a list of feature stories and requirements in the selected project.
09	`list_tasks`	Shows all granular tasks assigned within the current workspace for daily work.
10	`list_workspaces`	Provides an overview and list of every accessible project workspace you manage.

See It in Action

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

U List all my active workspaces on TAPD.

I've retrieved your TAPD workspaces. You have 4 active projects, including 'Cloud Infrastructure' and 'Mobile App V2'. Which one would you like to view items for?

U Create a new bug in workspace 'Mobile App V2' titled 'App crashes on splash screen'.

Done! I've created a new bug report 'App crashes on splash screen' in the 'Mobile App V2' workspace. I've assigned it a high priority by default. Would you like to add reproduction steps?

U Show me the iterations for project 'Cloud Infrastructure'.

I've listed the iterations for 'Cloud Infrastructure'. You have 3 iterations defined, including 'Q2 Milestone' (In Progress) and 'Security Patch v1.1' (Planned). Which one would you like to audit?

Frequently Asked Questions

01 **How do I start managing stories with the TAPD MCP?**

You start by asking your agent to list_workspaces to identify the correct project first. Then, you can use list_stories or create_story to manage requirements within that scope.

02 **Can I track bugs with the TAPD MCP?**

Yes. You can use list_bugs to see all existing defects and use create_bug to file a new report immediately, regardless of who owns the bug currently.

03 **Does the TAPD MCP help me manage my daily tasks?**

Absolutely. Use list_tasks or create_task to keep track of granular work items, ensuring nothing falls through the cracks between sprints.

04 **What is the difference between stories and tasks using the TAPD MCP?**

Stories are high-level requirements (the 'what' or 'why'). Tasks are the small, actionable steps needed to complete that story (the 'how').

05 **What if I want to know which team members are assigned to a feature?**

You can use list_members after specifying the workspace. This lets you see who is available for assignments or review.

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 `"tapd": { "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

TAPD 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 TAPD. 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	TAPD MCP
Server ID	`019d8487-a68e-736b-bc41-00570e1ce7a3`
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/tapd.