MCP SERVER | NO CODE | CLOUD HOSTED

BugHerd MCP for AI Agents

Track Visual Feedback and Manage Bug Reports

BugHerd connects your AI client directly to visual feedback and bug tracking workflows. Stop toggling between portals; use natural conversation to manage projects, triage new reports from clients, and update task priorities instantly. It lets you list all active development projects, retrieve detailed metadata, process incoming feedback queue items, and coordinate user involvement without ever logging into the BugHerd site.

F Quality Score 48.02/100

visual-feedback | bug-tracking | qa-testing | website-feedback | issue-management | collaboration

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

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.

BugHerd MCP

10 tools available
Cloud-hosted on Vinkius

BugHerd brings your entire visual feedback loop—from initial client report to final bug fix—into conversation with your AI agent. Instead of navigating through multiple tabs or copy-pasting links between a spreadsheet and a web portal, you talk to your AI client and it handles the rest. You can ask it to list all your current projects, get details on specific tasks, or even access the dedicated feedback queue to triage new reports from clients or team members.

It's about keeping your QA process inside your natural workflow. Whether you're working in a developer environment or writing documentation for product managers, your AI agent can pull core organizational settings and user directories right into your chat window. This integration becomes available through the Vinkius catalog, giving any MCP-compatible client immediate access to all these bug tracking tools.

You control the flow of feedback using natural language, making it dramatically faster to update task statuses or create new reports without leaving your current screen.

Core Capabilities

01 — Identify and manage projects

List all active development projects and retrieve detailed metadata for each one.

02 — Manage tasks and bugs

Get details on a specific task, list all tasks within a project, or update the status and priority of existing issues.

03 — Triage client feedback queue

Access and review new visual reports from your clients and team in the dedicated feedback queue.

04 — Create project components

Start a brand new project or create a new task/feedback report directly within your workflow.

05 — Coordinate organization details

Access the organization's core settings and retrieve a directory of all organization users.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and enter your BugHerd API Key.

02 Connect your preferred AI client (Claude, Cursor, etc.) to the MCP in your settings.

03 Use natural conversation with your agent. Ask it to list projects or update a task status, and it executes the action directly.

The bottom line is that you talk to your agent like talking to a coworker; it handles the complex API calls to BugHerd in the background.

Built For

This MCP is for anyone who spends time moving feedback data—from client emails, screenshots, and project boards—into actionable tasks. It targets web developers stuck jumping between dev tools and bug trackers, QA testers needing quick status checks, or project managers trying to keep track of multiple development streams.

Web Developer

Needs to quickly check task statuses or retrieve specific feedback details without manual logins to the BugHerd portal.

QA Tester

Must list all tasks in a project and update their status or priority when testing new features, keeping a detailed record of every bug found.

Project Manager

Needs to triage the dedicated feedback queue daily, assigning priorities and updating core organization information across multiple concurrent projects.

What Changes When You Connect

01 Instantly check project status or retrieve feedback details using the get_task and list_tasks tools, eliminating manual logins to your bug tracking portal.

02 Triage incoming client reports directly by calling list_feedback. You can review new items without ever leaving your primary workflow tool.

03 Maintain a clear record of all work streams by listing projects or creating brand new ones using create_project and get_project.

04 Keep project oversight tight: you can retrieve core organization settings via get_organization_info, keeping everyone on the same page about scope and rules.

05 Streamline communication by accessing user details through list_users, ensuring that every assigned bug has the right person involved.

Real-World Applications

Responding to a client's screenshot

A designer gets a high-res screenshot of a mobile layout issue. Instead of emailing it and waiting for someone to manually input it, they ask their agent: 'Can you create a new task in the Client Portal project with this description?' The agent uses create_task instantly.

Updating status across multiple teams

The QA tester finishes testing a module and needs to move three separate bugs. They ask: 'Update tasks A, B, and C in the Vinkius Redesign project to 'Done' and set priority to Low.' The agent uses update_task for all three.

Getting an overview of development scope

The Project Manager needs a quick list of every active site. They ask their agent: 'List all my BugHerd projects and give me the main URL for each.' The agent uses list_projects, providing immediate project oversight.

Onboarding a new team member

The manager needs to know who is on the current development roster. They ask: 'List all users in the organization.' The agent uses list_users and provides an immediate directory, speeding up coordination.

Patterns to Avoid

Using email threads for bug tracking

✗ AVOID

Sending a dozen emails about one issue, attaching screenshots to multiple threads, and manually consolidating status updates in a shared Google Sheet.

✓ INSTEAD

Instead of emailing, tell your agent: 'Use list_feedback to pull the latest reports from the client queue. Then use create_task to generate an official bug ticket for the development team.' It centralizes everything.

Ignoring task dependencies

✗ AVOID

Trying to update a bug status without knowing which project it belongs to, leading to incorrect data or failed updates.

✓ INSTEAD

Always start by listing projects (list_projects) or getting project details (get_project). This confirms the scope before you use any tool like update_task.

Overlooking user roles

✗ AVOID

Assuming a developer has access to all organizational settings or trying to assign tasks without confirming the person exists.

✓ INSTEAD

Always check who's available first. Use list_users to confirm team members, then use create_task specifying those verified users.

The Right Fit

Use this MCP if your bug tracking involves visual feedback (screenshots, layout issues) and you need that data processed alongside core project metadata. It's perfect for teams where QA or design handoffs are frequent.

Don't use this if your workflow is purely conceptual—for example, if you only track theoretical risks or market trends without a corresponding development task. If all you do is write specs in Notion and never update them in BugHerd, this MCP won't help.

If you need to integrate bug tracking into a larger system (like Jira or Asana), you might look at a dedicated workflow automation tool that connects multiple platforms, rather than just managing the data within BugHerd.

BugHerd and Visual QA Testing Workflow Management

Right now, finding a bug often involves an ugly, manual process. A designer spots something wrong on mobile, takes three screenshots, writes a detailed description in a separate email, and then someone has to manually copy that information into the project management board. It’s slow, it's fragmented, and half the context gets lost.

With this MCP, the workflow changes completely. You can ask your agent to pull new reports from the dedicated feedback queue using list_feedback. Your AI client reads the description, identifies the relevant project (get_project), and automatically drafts a task for you to approve. You get immediate, structured action items without any manual copy-pasting.

BugHerd and Project Oversight Coordination

Keeping track of multiple projects means constantly switching between dashboards just to see if the feature is ready. You have to remember which project has what URL, who is assigned, and what status it's currently in.

Now, you simply ask your agent: 'List all active development projects.' It provides the list right there, complete with metadata and URLs. This gives you instant, consolidated oversight over every single thing being built.

10 Tools in the BugHerd MCP for Issue Management

Use these tools to create, list, retrieve details on projects, tasks, and organization information using natural language prompts.

#	TOOL	DESCRIPTION
01	`create_project`	Builds a new project container within BugHerd for tracking related issues.
02	`create_task`	Generates a new task or feedback item in an existing project, ready for review.
03	`get_organization_info`	Fetches the core settings and details about your organization's account setup.
04	`get_project`	Retrieves all necessary information for a specific, existing project ID.
05	`get_task`	Grabs the full details and current status of any single task or bug report.
06	`list_feedback`	Pulls a list of all new, unassigned items directly from the client feedback queue.
07	`list_projects`	Generates a complete list of every project currently active in your BugHerd account.
08	`list_tasks`	Displays all tasks and bugs associated with a single, specified project.
09	`list_users`	Provides a directory listing of every user account tied to the organization.
10	`update_task`	Changes the status (e.g., To Do, Doing, Done) or updates the priority level on any existing task.

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 new payment gateway module?

Project: Payment Gateway Integration

- **Status:** Doing (Requires final QA sign-off)
- **Assigned To:** Jane Doe
- **Last Update:** 2 hours ago (Updated by system)
- **Priority:** High

Need to check the feedback queue for any blocker issues? I found 7 new items, including a reported failure on Chrome v120.

U I need to create a bug report for the mobile header alignment issue.

New Task Created Successfully

- **Project:** Vinkius Redesign (ID: proj_1)
- **Task Title:** Header misalignment on iPhone 14 Pro Max
- **Description:** The corner element overlaps the navigation bar when viewed in landscape mode.
- **Status:** To Do
- **Priority:** High

A new task has been logged and assigned to the development team.

U Can you list all my projects and tell me who is on them?

Active Projects Overview

PROJECT NAME	ID	STATUS
Vinkius Redesign	proj_1	Active
Client Portal	proj_2	Paused
Marketing Blog	proj_3	Complete

Users available across all projects include John Smith (role: PM), Jane Doe (role: QA), and Michael Chen (role: Dev).

Frequently Asked Questions

01 **How can BugHerd MCP help me manage client bug feedback?**

It lets you pull new reports from the dedicated queue using natural language. You don't have to jump into a separate client portal; your agent reads the latest submissions and helps you triage them right where you are working.

02 **Can I update task statuses without logging in?**

Yes, this MCP lets you communicate status changes directly. You just tell your agent which task needs updating and what its new status is (e.g., 'Doing' or 'Done'). It handles the API call for you.

03 **What kind of projects can I list with BugHerd MCP?**

You can list all your active development projects, getting names and essential metadata like unique IDs. This is useful for getting a high-level view of everything currently under development.

04 **Is BugHerd MCP only for big teams?**

No. It works for small teams too. You can use it to coordinate users and track specific tasks, making sure everyone knows who is responsible for what part of the project.

05 **Does using BugHerd MCP save time on onboarding new people?**

Yes. Instead of hunting through different documents, you can ask your agent to list all organization users immediately. It gives you a current directory and confirms roles for quick team coordination.

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 `"bugherd": { "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

BugHerd 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 BugHerd. 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	BugHerd MCP
Server ID	`019d7565-4eb4-7394-ab5c-cd27083548c3`
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/bugherd.