MCP SERVER | NO CODE | CLOUD HOSTED

BugSnag MCP for AI Agents

Monitor application stability and track error trends in software development

BugSnag connects your error monitoring into your AI agent, letting you track application stability without opening a dashboard. Your agent can list organizations or projects, inspect specific error groups, and retrieve deep details about individual error events using natural conversation. It lets engineers quickly diagnose issues by pulling real-time metrics and historical trends directly into their workflow.

F Quality Score 48.02/100

error-monitoring | stability-tracking | incident-response | full-stack-monitoring | debugging | application-performance

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

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.

BugSnag MCP

10 tools available
Cloud-hosted on Vinkius

Debugging complex software doesn't require manually hopping between dashboards and API calls. This MCP gives your AI client direct access to BugSnag, letting you manage application errors right where you work. Instead of searching through project settings or digging up specific error IDs, you just ask your agent a question about system health. It pulls everything—from listing all organizations to getting detailed statistics on an error group—and presents it instantly.

This capability means incident response gets faster and less painful. You can use natural language to get historical error trends, check collaboration status across projects, or even find out how many times a specific event has popped up in the last 24 hours. If you're already using Vinkius for other services, connecting BugSnag here makes your entire stack visible and actionable through one interface. It’s about getting immediate answers to operational questions that used to take five minutes of clicks.

Core Capabilities

01 — Check organizational visibility

List all the organizations you have access to, giving you a high-level view of your entire tech portfolio.

02 — Identify project errors and groups

List and inspect error groups for any specific project, showing details like severity levels and how often those errors occur.

03 — Diagnose individual incidents

Retrieve full records of single error events to pinpoint exactly when and where a failure happened.

04 — Review system health over time

Get detailed statistics on error trends for a project, helping you monitor application stability across releases.

05 — Manage team context

Access directory information about collaborators and the current release stages to keep all teams aligned.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP on Vinkius and enter your BugSnag Personal Auth Token.

02 Next, tell your AI client what you need—for example, 'What are the top three error groups for my production web app?'

03 Your agent sends that query through the connection, retrieves the data from BugSnag, and presents a clean summary of the error details or statistics back to you.

The bottom line is, you talk to your AI client about bugs instead of navigating complex dashboards.

Built For

This MCP is built for people whose job involves keeping software running. If you're an SRE staring at a dashboard at 3 AM wondering why the error rate spiked, or a developer who needs to stop clicking through five different tabs just to find one specific event ID, this tool saves you time.

Site Reliability Engineer (SRE)

Monitors stability trends and reviews release stage health straight from their workflow tools. They need to know if the latest deployment introduced a new error group.

Software Developer

Needs quick access to specific error event details or counts without manually searching through dashboards, allowing them to jump faster into code fixes.

DevOps Engineer

Uses the MCP to get high-level error statistics and metadata for incident response, ensuring team coordination is always accurate.

What Changes When You Connect

01 Pinpoint root causes faster. Instead of just knowing an error happened, you can retrieve the full event details using `get_event` to debug exactly why it failed.

02 Get a complete picture of your infrastructure. You don't have to guess where to look; simply use `list_organizations` and `list_projects` to map out your entire tech stack's visibility.

03 Keep stakeholders informed easily. Use `list_collaborators` to quickly check who is on the team and what release stages they are working with, keeping everyone aligned during an incident.

04 Understand performance shifts. By calling `get_project_stats`, you track error trends over time, helping you prove whether a recent code change actually improved or worsened stability.

05 Consolidate your knowledge. You can use natural language to compile data from multiple calls—like combining results from `list_errors` and `list_events`—without leaving the chat window.

Real-World Applications

Investigating a spike in production errors

A developer notices error reports spiking after deployment. They ask their agent to 'Show me all active error groups for the web dashboard.' The MCP uses `list_errors` and `get_error`, immediately identifying the high-severity group, allowing them to pull the specific project details using `get_project` and start debugging.

Onboarding a new team member

A PM needs an overview of all system health. They ask their agent to 'List every organization and project we monitor.' The MCP runs `list_organizations` followed by `list_projects`, generating a clean inventory that shows the full scope of monitoring coverage.

Debugging flaky API calls

An SRE suspects a specific, intermittent bug. They ask their agent to 'Get details for event ID XYZ.' The MCP uses `get_event`, providing metadata and occurrence counts instantly, letting the SRE confirm if it was a one-off failure or a systemic issue.

Assessing post-release impact

A manager wants to know if the latest feature release impacted stability. They ask their agent for 'Error trends and statistics on the mobile app.' The MCP runs `get_project_stats`, delivering a clear, actionable graph showing performance changes over time.

Patterns to Avoid

Manual Dashboard Hunting

✗ AVOID

A developer has to navigate through the BugSnag UI, select a project from a dropdown, then filter by error type, and finally copy an event ID. This process takes several minutes of clicking.

✓ INSTEAD

Instead, ask your agent directly: 'What are the most common errors in my iOS App?' The MCP handles the `list_projects` and `list_errors` calls automatically, giving you a summarized list without any manual navigation.

Forgetting Context

✗ AVOID

Trying to check error details but forgetting which organization or project the issue belongs to, leading to generic search results.

✓ INSTEAD

Always start by asking your agent to 'List all my organizations.' This gives you context via `list_organizations`, ensuring every subsequent query —like getting an error group's details using `get_error` —is scoped correctly.

Misinterpreting Severity

✗ AVOID

Assuming a 'Warning' level event means everything is fine, when it might actually indicate a critical performance degradation.

✓ INSTEAD

Use the MCP to get specific project stats (`get_project_stats`) and ask your agent: 'What are the severity trends for our web dashboard?' This shows you if warnings have been trending up over multiple release stages.

The Right Fit

Use this MCP if your primary job involves reacting to software failures, diagnosing intermittent bugs, or monitoring application health across multiple services. If you frequently need to compare error groups against historical data, getting project statistics, or checking the status of specific releases, this is a must-have. Don't use it if all you do is write code that never breaks; in that case, you don't need error monitoring at all. Also, don't rely on it for business logic—it only reads data. For coordinating people or tasks outside of technical failure reporting, consider dedicated communication tools instead.

BugSnag MCP: Streamlining Error Group Monitoring with AI Agents

Right now, when a critical error hits production, the first thing you do is open BugSnag. You navigate to the right project, filter by time frame, and then manually check if an error group has spiked in frequency or severity. This involves clicking through several tabs just to determine the scope of the problem.

With this MCP, you simply ask your agent: 'Show me all critical errors for the last hour.' The system handles listing projects, fetching the relevant groups via `list_errors`, and presenting a clean summary right in your chat. You get immediate context without leaving your workspace.

BugSnag MCP: Tracking Stability Across Software Releases

Manually comparing stability between release stages is a nightmare. You have to check the 'staging' environment, then switch to 'pre-prod,' and manually pull stats for each one—all while hoping you don't miss a critical trend.

Now, ask your agent: 'Compare error trends between staging and production.' The MCP pulls `get_project_stats` and synthesizes the data, giving you an instant, side-by-side view of stability across all configured release stages.

10 Tools for BugSnag: Event Statistics and Error Group Management

Use these tools to get comprehensive insights into your application's health, from listing all projects to retrieving detailed error events.

#	TOOL	DESCRIPTION
01	`get_error`	Pulls specific details about a defined group of errors (an error class).
02	`get_event`	Retrieves full, unique details for one single instance of an error event.
03	`get_project`	Gets key information about a specific software project within the system.
04	`get_project_stats`	Calculates and provides error trends and overall statistics for an entire project over time.
05	`list_collaborators`	Displays a directory of team members who work in the same organization.
06	`list_errors`	Generates a comprehensive list of all error groups found for a given project.
07	`list_events`	Creates a navigable list of individual, historical error events tied to a specific project.
08	`list_organizations`	Provides an inventory of all BugSnag organizations you have access to monitor.
09	`list_projects`	Lists every software project configured within a specific organization.
10	`list_release_stages`	Shows the various release stages that are currently set up and monitored for a project.

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 projects in BugSnag for organization vinkius corp.

Vinkius Corp Projects

The following 4 projects are currently monitored:

- 🌐 **Web Dashboard**: ID: proj_1 (Last Error: ReferenceError)
- 📱 **iOS App**: ID: proj_2 (Last Error: NetworkError)
- 🤖 **API Gateway**: ID: proj_3 (Last Error: TypeError)
- ⚙️ **Auth Service**: ID: proj_4 (Last Error: NullPointer)

U What are the details for error group 'err_99283'?

Error Group Details: err_99283

- **Name**: ReferenceError
- **Severity**: Error (High)
- **Occurrences (Last 24h)**: 150 times
- **Affected Users**: ~45 users in production.
- **Description**: This error happens when an object expected to be defined isn't found. It usually relates to outdated API endpoints.

U Show the top 5 error events for 'iOS App'.

Top 5 Recent Failures (iOS App)

EVENT ID	TYPE	SEVERITY	TIME AGO
ev_102	NetworkError	Warning	1h ago
ev_987	TypeError	Error	4h ago
ev_551	ReferenceError	Critical	6h ago
...	...	...	...

Note: The most frequent event is `TypeError` (32 occurrences in the last week).

Frequently Asked Questions

01 **How can BugSnag MCP help me debug production errors?**

Your agent lets you query real-time and historical error data directly. You can ask for specific event details or list all current error groups, speeding up diagnosis without needing to open the main dashboard.

02 **Does BugSnag MCP help with release version tracking?**

Yes. It lets you check which software projects are configured across different release stages (like staging or production). This is crucial for understanding if an error started appearing after a specific deployment.

03 **What kind of reports can I get from the BugSnag MCP?**

You can pull various reports, including comprehensive statistics on error trends over time and deep dives into individual error groups. This gives you both high-level metrics and low-level diagnostic data.

04 **Is this only for small errors or major outages?**

It handles both. You can get a general overview of all your organizations, or drill down to retrieve the metadata for a single, highly critical error event that requires immediate attention.

05 **Can BugSnag MCP tell me if two projects are related?**

It helps you map out relationships by listing all available projects within an organization and checking which collaborators work across multiple services. This ensures your team is aligned on the scope of monitoring.

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 `"bugsnag": { "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

BugSnag 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 BugSnag. 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	BugSnag MCP
Server ID	`019d7565-6ab9-73c4-987b-f054f3d60a55`
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/bugsnag.