MCP SERVER | NO CODE | CLOUD HOSTED

GitBook MCP

Control your entire knowledge base from conversation.

GitBook MCP connects your AI agent directly into your technical documentation platform, giving you full control over knowledge bases and docs-as-code workflows. You can list every organization and space, search content across entire namespaces, audit metadata for visibility rules, and retrieve specific page content without ever opening the GitBook UI.

A+ Quality Score 100/100

documentation | docs-as-code | technical-writing | knowledge-base | collaboration

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

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.

GitBook MCP

8 tools available
Cloud-hosted on Vinkius

Your agent takes control of all your technical documentation. Instead of navigating through menus or copying text from multiple sources, you talk to your knowledge base directly. This MCP lets you list every organization and space mapped in GitBook, giving you an immediate overview of your entire product documentation structure. You can audit specific spaces for access rules or read content pages wholesale just by asking questions. Need to know how different parts of the company organize their docs? Use the collection management tools to map out that hierarchy. If you need to find a single piece of information buried deep in hundreds of pages, run a cross-page search using natural language queries. When you connect this MCP via Vinkius, your agent can treat documentation like any other data source—a structured resource ready for immediate use.

Core Capabilities

01 — Map organizational structure

List all organizations and spaces within your GitBook profile to understand the full scope of your documentation.

02 — Discover page content

Read entire document pages or traverse a space's hierarchy to extract technical information flawlessly.

03 — Search across namespaces

Execute natural language searches inside your GitBook, pulling relevant snippets from multiple documentation spaces.

04 — Audit structural metadata

Fetch detailed information about specific document spaces to verify their visibility and access rules.

05 — Manage content groupings

List collections that group multiple documentation spaces, providing a high-level view of knowledge distribution across your company.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and generate an API Token in your GitBook developer settings.

02 Enter the generated GitBook API token into your AI client's connection settings.

03 Ask your agent a question about your documentation, like 'Show me all spaces under the Marketing organization,' and it executes the necessary calls.

The bottom line is you get direct command access to read, list, and search every part of your GitBook knowledge base using natural language prompts.

Built For

This MCP is for the technical writers who spend hours clicking through documentation trees. It's for product managers needing a real-time audit of knowledge distribution, and customer support teams that need to find complex answers instantly without reading manuals.

Technical Writer

Uses this MCP to verify content structure or retrieve specific page IDs quickly, allowing them to debug documentation integrations directly from their agent.

Product Manager

Audits collections and spaces to analyze knowledge gaps across different product lines without manually checking every document silo.

Customer Support Lead

Uses the agent to perform cross-page searches, compiling accurate technical answers from all product documentation simultaneously for rapid issue resolution.

What Changes When You Connect

01 Skip the clicks. Instead of navigating through a dozen tabs to find documentation, you ask your agent for information and get it instantly using cross-page search operations.

02 Audit your knowledge structure easily. Use tools like list_collections or get_space to map out exactly how different product documentations are organized across multiple organizations.

03 Eliminate manual content retrieval. The get_page tool pulls the full, raw text from any page ID, letting you analyze and process the content without needing a copy/paste workflow.

04 Verify permissions instantly. By using tools like get_me or get_space, you can check detailed metadata about spaces to confirm visibility and structural configuration rules before writing anything.

05 Handle complex hierarchies. You don't just see pages; you see the entire page hierarchy for a space, helping you understand how content sections relate to one another.

Real-World Applications

Mapping out product knowledge gaps

A Product Manager needs to know if 'Advanced Security' is documented consistently across three different product lines. They run list_collections and then use get_space on each resulting collection to audit the overall documentation scope, identifying missing sections before a release.

Resolving complex support tickets

A Customer Support Lead gets a difficult question about API authentication. They ask their agent to search_content across all 'API Reference' spaces and get the most relevant page content, delivering an immediate answer instead of searching multiple manuals.

Debugging documentation integrations

A Developer needs to verify if a new section is correctly placed in the User Guide. They use list_pages first, then get_space to check the metadata, ensuring the structure matches the required schema before writing any code.

Getting an overview of all company docs

A Technical Writer needs a quick inventory of every documentation area. They start by using list_organizations, followed immediately by list_spaces to get a comprehensive map of the entire internal knowledge base.

Patterns to Avoid

Treating GitBook like a simple file cabinet

✗ AVOID

Assuming you just need to read one page. You might only call get_page on one document, missing the broader context of that entire section or collection.

✓ INSTEAD

Always start by using list_spaces and then getting the space details with get_space. This gives you the structural map first, ensuring you know which collections are relevant before focusing on a single page.

Manually listing every document piece

✗ AVOID

Trying to remember if all documentation spaces have been updated or audited for security rules, leading to incomplete knowledge maps.

✓ INSTEAD

Use the dedicated get_space tool. It fetches detailed metadata about a specific space, verifying access rules and structural configurations in one call.

Searching without context

✗ AVOID

Simply inputting keywords into an agent without limiting the search scope. The results are too broad, mixing relevant documentation with unrelated drafts.

✓ INSTEAD

Always narrow your focus by first using list_spaces to target a specific space (e.g., 'API Reference') and then running search_content only within that identified namespace.

The Right Fit

Use this MCP if your primary need is controlling the structure, content, and metadata of an existing GitBook knowledge base through API calls. You need to list organizations or spaces, audit visibility rules, or run cross-page searches across multiple namespaces. Don't use it if you just want to draft documentation; that requires a writing tool. Similarly, don't use it if your goal is simply to manage user accounts—use the get_me tool only for verification purposes. If you are trying to connect GitBook data into an external application like Notion or Confluence, this MCP won't help because its job is purely retrieval and structural analysis of the source platform itself.

The headache of navigating massive documentation platforms.

Today, finding a single answer across your company's technical docs means clicking through multiple portals. You check the API section, then jump to the user guide, and finally consult the release notes—copying small snippets into a temporary document until you piece together what you need.

With this MCP, that friction vanishes. Your agent performs deep dives for you, pulling data across entire namespaces. It’s not about reading pages; it's about getting the specific facts from anywhere in your knowledge base.

GitBook: Structural Clarity with GitBook MCP

You stop relying on remembering where every single document lives. You no longer have to manually check if a collection is correctly grouping spaces or verify the access rules for an entire product line.

What changes now is that your knowledge base becomes a queryable data source, giving you immediate clarity over its structure and content depth.

GitBook: 8 Documentation Management Tools

These tools give your AI client granular control over every aspect of your GitBook knowledge base, from listing top-level organizations to retrieving specific page content.

#	TOOL	DESCRIPTION
01	`list_organizations`	Retrieves a list of every organization account connected to your GitBook profile.
02	`list_spaces`	Displays all individual documentation spaces belonging to a specified organization.
03	`get_space`	Retrieves detailed structural and metadata information about a specific document space.
04	`list_pages`	Lists the titles and hierarchy of pages contained within a single documentation space.
05	`get_page`	Pulls the complete, raw text content from a specified page ID.
06	`search_content`	Performs a natural language search across multiple pages within a designated space to find relevant snippets.
07	`list_collections`	Lists high-level collections that group related documentation spaces together.
08	`get_me`	Retrieves basic metadata about the authenticated user, including their name and email address.

See It in Action

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

U List all spaces in organization 'org_123'

Retrieving spaces... I found 3 documentation environments in 'org_123': 'API Reference', 'User Guide', and 'Release Notes'. Which one would you like to explore for pages?

U Search my GitBook for 'authentication flow'

Searching content... I found several matches in the 'API Reference' space. The most relevant is a page titled 'Auth Overview' (ID: page_abc). Would you like me to retrieve the full content of that page?

U Show me the page hierarchy for space 'User-Guide'

Retrieving hierarchy... The 'User Guide' space has a main 'Getting Started' section, followed by 'Dashboard Basics' and 'Advanced Settings'. Each section has multiple nested pages. Would you like the ID for any specific page?

Frequently Asked Questions

01 **How do I list all documentation spaces using the GitBook MCP?**

You use the list_spaces tool. After listing organizations with list_organizations, calling list_spaces lets you see every individual space contained within that organization.

02 **Can I read a page without knowing its ID? (GitBook MCP)**

No, direct retrieval requires the page ID. However, you can use list_pages to get the hierarchy and find the necessary IDs first.

03 **Does GitBook MCP allow me to search across all my product lines?**

Yes, you run a search using search_content on the space that contains multiple related products. This allows cross-page searching within your specified namespace.

04 **What if I need to check who is viewing the documentation? (GitBook MCP)**

You can retrieve basic profile metadata using get_me, which gives you authenticated user information like name and email for context verification.

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 `"gitbook": { "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

GitBook 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 GitBook. 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	GitBook MCP
Server ID	`019d75a5-1d05-7091-8728-1ac126b96392`
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/gitbook.