MCP SERVER | NO CODE | CLOUD HOSTED

Feishu Bitable MCP

Manage and query structured database records naturally.

Feishu Bitable MCP connects your AI agent directly to Lark's multi-dimensional table platform. It lets you manage complex databases, track records, and orchestrate data structures using only natural conversation. Instead of clicking through dozens of tabs, you simply tell your agent what you need—like listing all tables or updating specific fields—and get the result instantly.

A+ Quality Score 100/100

database-management | no-code | record-tracking | collaborative-data | schema-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/feishu-bitable

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.

Feishu Bitable MCP

10 tools available
Cloud-hosted on Vinkius

Managing a database shouldn't require navigating a web interface. With this MCP, your AI client treats Feishu Bitable like a conversational API. You can ask your agent to perform actions across multiple bases and tables without knowing where the data lives or how deep its structure goes. For example, you don't have to remember which table holds project status vs. resource allocation; you just tell your agent to get base information, and it finds what you need.

This capability means that whether you are tracking inventory, managing a client roster, or running operational reports, all data operations happen in plain language. Your agent handles the complexity—it can list fields, search through records with filters, batch create new entries, or even update existing information using `update_record`. It's like having an expert data administrator who lives inside your chat window. This kind of deep integration is what Vinkius makes possible, turning static databases into dynamic parts of your daily workflow.

Core Capabilities

01 — Discovering Data Structures

You can ask the agent to list all available tables in a base and retrieve field schemas so you know exactly what data points exist.

02 — Querying and Finding Records

The agent searches records using custom filter expressions, allowing you to pinpoint specific entries across large datasets.

03 — Modifying Database Data

You can instruct the system to create new batches of records, update existing entries, or delete data directly through conversation.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and enter your specific Feishu/Lark App ID and App Secret.

02 Next, connect the MCP to any compatible client—like Claude or Cursor. This establishes a secure connection between your agent and Bitable.

03 Finally, talk to your AI client naturally. Tell it what data you need, and your agent will run the necessary operations against your bases.

The bottom line is: once connected, you use natural language prompts instead of complex UI navigation to manage your entire database setup.

Built For

This MCP is for anyone whose job involves maintaining structured data across multiple platforms. If you're tired of manually checking dashboards or running reports just to find a single piece of information, this tool saves your afternoon.

Operations Analyst

Needs to monitor consistency across dozens of project bases and use the agent to `get_base_info` before running any large data operations.

Project Manager

Requires the ability to quickly check milestones by asking the agent to search records, rather than building a complex filter query in the UI.

Data Entry Specialist

Uses the tool's conversational abilities to `create_records` or perform batch updates without ever needing to open the base editor.

What Changes When You Connect

01 Stop building complex filters. Instead of manually setting up advanced search criteria, just ask your agent to `search_records` for what you need, and it handles the syntax.

02 Eliminate tedious data entry. You can tell your agent to `create_records` in a batch process, populating dozens of entries without ever leaving your chat window.

03 Get full context instantly. Need to know which tables exist? Use `list_tables` first. The MCP gives you the structural overview before you try to read any data.

04 Never lose track of schema changes again. If you're unsure what fields a table has, run `list_fields`. It provides an immediate audit of the structure.

05 Audit and maintain consistency using `get_base_info` or `get_record_details`. You can confirm data integrity simply by asking your agent for details on specific bases.

Real-World Applications

Auditing Old Project Data

A project lead needs to know the status of all clients in a base created years ago. Instead of navigating through multiple views and running manual queries, they ask their agent to `list_tables` first, then use `search_records` across the correct table for records matching 'Status: Pending Review'. The agent returns only the relevant IDs.

Onboarding New Team Members

An operations manager needs to ensure a new hire's data is correctly entered across several project tables. They prompt their agent, asking it to `create_records` for all standard fields in the 'Employee Info' table. The agent handles the required formatting and validation.

Debugging Data Discrepancies

A data analyst spots a potential error. They use `get_record_details` on a suspicious record ID to pull all field values instantly, allowing them to compare it against the expected schema retrieved via `list_fields`.

Quickly Updating Project Scope

A project manager has just received scope changes for 15 clients. Instead of opening 15 different records and manually changing fields, they instruct their agent to `update_record` for all relevant client IDs with the new scope details.

Patterns to Avoid

Manual Data Querying

✗ AVOID

The user spends 15 minutes clicking through tabs, filtering views, and running multiple manual searches in the web interface to piece together a small dataset.

✓ INSTEAD

Instead of navigating the UI, ask your agent to `search_records` directly. Provide the table name and filter criteria in one prompt. The MCP handles the complex query generation.

Guessing Field Names

✗ AVOID

A user assumes a field is named 'Client Status' but it's actually called 'Cust_Status'. They waste time trying to find data that doesn't exist.

✓ INSTEAD

Before attempting any operation, ask the agent to `list_fields` for that table. This confirms all available field names and structures before you write a single prompt.

Trying to Update Data in Bulk Manually

✗ AVOID

A user has 50 records that need one single field updated, forcing them to open 50 individual pages and manually change the data.

✓ INSTEAD

Use `update_record`. Pass the list of record IDs and the single field/value pair to your agent. It executes the update across all items in a single command.

The Right Fit

Use this MCP if your core problem is managing structured, relational data within Feishu Bitable's environment. The power here comes from treating the database itself as an actionable endpoint; you are talking to the *structure*, not just viewing it. You need conversational control over creation, reading, updating, and deleting records.

Don't use this if your goal is merely text generation or summarizing unstructured documents (use a pure LLM connector instead). If all you need is to read data from a single, simple spreadsheet without any complex filtering or writing back, a basic endpoint connector might suffice. But for anything that requires knowing the *schema*—if you must first `list_fields` before you can `create_records`—this MCP is your go-to tool.

The Pain of Database Navigation

Today, if you need to update a client record or pull a specific report, you open the Bitable base. You navigate to the correct table, then maybe click through multiple views until you find the right segment. Then, if you want to change something, you have to manually edit it and hit save. If you need data for 50 clients, you're copying values into an external spreadsheet—it’s a messy, slow process.

With this MCP, those manual clicks vanish. You just tell your agent what needs fixing or tracking. For instance, instead of opening the base, asking it to `list_tables` confirms its existence and structure. Then, you ask it to search for records using specific criteria, pulling the exact data points into your conversation instantly.

Feishu Bitable MCP: Data Control via Conversation

The manual steps that disappear are the initial discovery phase and the execution. You don't have to manually verify which table holds 'Inventory Count' or 'Client Contact Info'; you just ask your agent to `list_fields` first, confirming the exact column name.

The difference is control. It’s not just about getting data; it’s about commanding the database to act on your behalf, allowing you to manage record entry and updates through natural conversation.

Feishu Bitable: 10 Tools for Data Management

These tools allow your agent to perform every core database operation—from listing table schemas to updating individual records—all through natural language prompts.

#	TOOL	DESCRIPTION
01	`create_records`	Batch creates multiple new entries in a specified table with defined field values.
02	`delete_record`	Permanently removes an identified record from the target table.
03	`get_base_info`	Retrieves high-level metadata and structural information about an entire Bitable base.
04	`get_record_details`	Fetches all the specific, detailed data points for a single record ID.
05	`list_fields`	Lists every field available within a given table to confirm structure and names.
06	`list_records`	Retrieves all records currently stored in a specific, non-filtered table view.
07	`list_tables`	Lists every individual table that exists within the selected Bitable base app.
08	`list_views`	Shows all defined virtual views for a table, helping you understand how data is segmented.
09	`search_records`	Searches and returns records that match specific criteria or filter expressions across the dataset.
10	`update_record`	Modifies one or more fields in an existing record using its unique ID.

See It in Action

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

U List all tables in Bitable base 'AppbcbWCzen6D8dezhoCH2RpMAh'.

I've retrieved the tables for the specified Bitable base. There are 3 tables: 'Inventory', 'Suppliers', and 'Orders'. Which one would you like to view records for?

U Add a new record to the 'Inventory' table with fields: Name='MacBook Pro', Quantity=5.

Done! I've added the new record to the 'Inventory' table. The entry for 'MacBook Pro' with a quantity of 5 has been successfully created.

U Search for records in table 'tblsRc9GRRXKqhvW' where 'Status' equals 'Shipped'.

I've performed the search and found 8 records matching your criteria. Would you like me to list their details or summarize the results?

Frequently Asked Questions

01 **How do I start using Feishu Bitable MCP to read my data?**

You first need to subscribe to the MCP and enter your App ID/Secret. Once connected, you can ask your agent to `list_tables` to see what bases are available.

02 **Can I bulk update records using Feishu Bitable MCP?**

Yes, you use the `update_record` tool. You provide a list of record IDs and the specific field/value changes, and your agent handles the batch modification.

03 **Is Feishu Bitable MCP only for reading data?**

No. While it excels at retrieval, it also allows you to write data back into Bitable using `create_records` or `update_record`, making it a full lifecycle tool.

04 **What if I need to find a record that doesn't fit simple filters?**

Use the `search_records` tool. It supports complex filter expressions, allowing you to search across multiple conditions simultaneously without needing to know the underlying query language.

05 **Does Feishu Bitable MCP help with data structure planning?**

Yes. You can use `list_fields` and `get_base_info` to audit your current schema, helping you confirm if all necessary fields exist before starting a project.

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 `"feishu-bitable": { "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

Feishu Bitable 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 Feishu Bitable. 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	Feishu Bitable MCP
Server ID	`019d843a-b99b-73b3-bb6b-66087ba91171`
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/feishu-bitable.