MCP SERVER | NO CODE | CLOUD HOSTED

ToolJet MCP

Query, insert & automate data tasks via AI chat.

ToolJet allows your AI agent to become a backend developer for your own systems. It connects directly to internal databases so you can run SQL queries, insert records, or trigger complex business workflows—all through natural language conversation. Stop leaving the chat window to manage data; start doing it right where you are.

A+ Quality Score 100/100

low-code | internal-tools | sql-query | workflow-automation | database-management | business-logic

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

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.

ToolJet MCP

4 tools available
Cloud-hosted on Vinkius

This MCP lets any AI client connect to your private ToolJet database and automation layer. You don't have to write code or navigate a clunky UI just to check an order status or update a customer record. Instead, you tell your agent what you need—like 'Show me all pending orders for last week'—and it handles the structured interaction with your data tables. The system executes the query and hands back clean results instantly. It also lets you kick off full business processes by triggering specific workflows, passing custom details like user IDs or product codes when needed. Because Vinkius manages this catalog connection, you get access to all these capabilities from one place, letting your agent actually perform actions on behalf of your business.

Core Capabilities

01 — Discover available data tables

You can ask the agent what database tables exist without knowing their names.

02 — Run selective data queries

The agent executes SQL SELECT statements to filter, sort, and retrieve specific records from your tables.

03 — Add new structured data entries

You can instruct the agent to insert brand-new rows of information into a database table using JSON details.

04 — Automate business processes

The agent triggers complex, multi-step workflows within ToolJet by sending a simple webhook call.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your unique ToolJet API Key (and optional self-host URL).

02 Next, tell your AI agent what you need done—for example, 'Get me all users who signed up last month' or 'Reset the billing status for user 123'.

03 The system converts that request into a database query or workflow trigger and returns the resulting data or confirmation message to your chat.

The bottom line is you get structured, automated access to your business logic without writing boilerplate code.

Built For

This MCP is essential for operations teams and data analysts who spend too much time jumping between spreadsheets, dashboards, and database UIs just to answer simple questions or update records. It's for the developers tired of writing repetitive boilerplate code.

Data Analyst

Runs ad-hoc queries on specific tables to verify data integrity or pull subsets of information needed for reports, without needing a BI tool.

Operations Manager

Updates customer records, flags processes, or starts automated business procedures using natural language instead of clicking through complex web dashboards.

Developer

Tests database interactions or triggers CI/CD workflows directly from the chat interface while working in their IDE.

What Changes When You Connect

01 Stop switching contexts. You can run complex reports or check specific records using `query_table` right inside your chat window, eliminating the need to open a separate SQL client.

02 Handle routine data entry immediately. Instead of filling out a web form, you just ask your agent to use `insert_row`, providing structured JSON details for instant record creation.

03 Automate entire processes from conversation. Use `trigger_workflow` to kick off complex backend logic—like sending an onboarding email or updating inventory—with a single command.

04 Quickly map your data structure using `list_tables`. This lets you understand exactly what's in the database before writing any query, saving hours of guesswork.

05 The whole thing works natively across multiple platforms. You connect once to Vinkius and get this powerful database access no matter which AI client you prefer.

Real-World Applications

Investigating a delayed payment status

An operations manager needs to know why an order is stuck. They ask their agent, 'What are the details for pending orders in organization X?' The agent uses `query_table` and returns exactly two records with notes on the delay.

Onboarding a new user's account data

A developer needs to manually create test accounts. They tell their agent, 'Insert a new user for Jane Doe with department Marketing.' The agent uses `insert_row` and guarantees the record is structured correctly.

Running an end-of-month reconciliation

A data analyst needs to check if all regional dashboards are up to date. They ask their agent, 'List all available tables' first, then run a targeted `query_table` across inventory and orders to spot discrepancies.

Resetting a client's access after departure

An ops team member needs to execute a complex off-boarding procedure. They ask their agent to 'Trigger the full user deactivation workflow for ID 501.' The agent uses `trigger_workflow` and confirms the process started successfully.

Patterns to Avoid

Treating it like a general chat bot

✗ AVOID

Asking, 'Hey, tell me about your database,' and expecting conversational knowledge instead of structured data. The agent can't answer vague questions.

✓ INSTEAD

Always ask for specific actions using the tools. Instead of asking generally, use `list_tables` first to understand structure, then follow up with a precise request like 'Run a query on the orders table where status is pending'.

Trying to write code manually

✗ AVOID

Writing out a complex SQL SELECT statement in your prompt when you just need simple filtering. This is verbose and error-prone.

✓ INSTEAD

Let the agent use `query_table`. You only need to describe the desired outcome naturally, and the MCP handles turning that into optimized SQL.

Forgetting the workflow step

✗ AVOID

Thinking data retrieval (reading) is enough. Missing the point that sometimes you have to kick off an action (writing/automating) after finding data.

✓ INSTEAD

Remember that if your goal is a change in state—like sending an alert or updating inventory—you need to use `trigger_workflow` after confirming the necessary details.

The Right Fit

Use this MCP if your primary bottleneck is accessing, modifying, or acting on structured data defined by existing business logic. If you are in a role that needs to move from 'What does my database contain?' to 'I need the system to do X based on Y data,' this is what you need. This tool's strength is translating natural language intent into precise backend operations using `query_table` for reading, and both `insert_row` and `trigger_workflow` for writing or automating.

Don't use this if you just want the AI to summarize a document, brainstorm ideas, or generate creative text. For those tasks, a general-purpose language model is fine. You need this when your conversation needs to result in a guaranteed change of state within an external system.

The Dashboard Fatigue

Today, if you need to check data or update records, the process involves jumping through hoops. You open the database UI to list tables, then switch to a query editor and write SQL (or run a pre-built report). If that doesn't work, you copy key IDs into a separate spreadsheet, manually calculate what needs changing, and finally go back to another dashboard just to hit 'Save.'

With this MCP, the entire sequence collapses. You tell your agent exactly what you need—for instance, 'Find all users in Chicago who haven't logged in for 90 days.' The agent handles the `query_table` instantly and gives you the list. No clicks, no context switching, just the answer.

ToolJet MCP: Database & Workflow Ops

You eliminate manual data entry entirely. Instead of opening a form to record new leads or product updates, you simply prompt your agent to use `insert_row`, giving it the required details in conversation. You also skip running repetitive

Your AI client now has hands—it can read, write, and act on your business data immediately. It’s a direct bridge from thought to execution.

scripts by having the agent call `trigger_workflow` for complex tasks.

ToolJet: Database & Workflow Ops (4 Tools)

Use these four specific tools—list tables, query, insert rows, and trigger workflows—to perform structured operations on your internal data directly through your AI client.

#	TOOL	DESCRIPTION
01	`insert_row`	Adds a completely new row of data into any specified ToolJet Database table.
02	`list_tables`	Retrieves a list of all available tables within the connected ToolJet database for discovery.
03	`query_table`	Executes specific SQL SELECT queries against your data to retrieve filtered records and generate insights.
04	`trigger_workflow`	Initiates a complex ToolJet Workflow process by sending a webhook payload, starting automated business logic.

See It in Action

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

U List all the tables available in my ToolJet Database.

I've retrieved the tables for your workspace. You have 3 tables: 'inventory', 'orders', and 'customers'.

U Run a query to find all orders with a status of 'pending' in organization 'org_123'.

Executing `query_table` ... I found 2 pending orders: Order #501 (Total: $120) and Order #505 (Total: $45).

U Trigger the workflow 'wf_987' with the payload {"user_id": "123", "action": "reset"}.

The workflow 'wf_987' has been triggered successfully. The response from ToolJet indicates the process has started.

Frequently Asked Questions

01 **How does the ToolJet MCP handle sensitive data?**

The connection requires you to provide an API key for controlled access. The agent only executes commands (like `query_table` or `insert_row`) based on your explicit prompt, ensuring all operations are tracked and constrained by your setup.

02 **Does the ToolJet MCP support complex filters?**

Yes. When you use `query_table`, the agent generates structured SQL that supports advanced filtering, joining tables (if necessary), and sorting data to get precise results.

03 **Can I use this MCP if my ToolJet instance is self-hosted?**

Absolutely. The setup process allows you to provide an optional Host URL, making it compatible with your private or self-managed instances.

04 **What types of workflows can the ToolJet MCP trigger?**

The `trigger_workflow` tool executes any workflow defined within your ToolJet environment. These processes are limited only by the logic and steps you've built into ToolJet itself.

05 **Is the ToolJet MCP just for reading data?**

No, it’s far more powerful than that. You can read using `query_table`, write using `insert_row`, and automate processes using `trigger_workflow`.

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 `"tooljet": { "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

ToolJet 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 ToolJet. 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	ToolJet MCP
Server ID	`019e38fd-a9aa-7186-92dc-5595bc64f461`
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/tooljet.