MCP SERVER | NO CODE | CLOUD HOSTED

IATA Developer Portal MCP

Audit Global Aviation Codes via Natural Chat

IATA Developer Portal provides immediate access to official global aviation standards, letting your AI agent audit airports, airlines, and aircraft data without manual lookup. Need to verify a flight path or build a travel app? This MCP lets you search for airport codes, check airline identifiers against the master list, and retrieve detailed metadata using natural conversation.

A+ Quality Score 100/100

aviation-standards | airport-codes | airline-identifiers | logistics-auditing | industry-data | reference-data

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/iata-developer-portal

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.

IATA Developer Portal MCP

6 tools available
Cloud-hosted on Vinkius

Trying to research global logistics means wading through massive industry manuals filled with acronyms and version numbers. You shouldn't have to be an aviation expert just to write code or plan a route.

The IATA Developer Portal connects your AI agent directly to the world's authoritative source for aviation data. Instead of copy-pasting codes into fragmented spreadsheets, you ask your agent a question—like 'What are the details for Heathrow?'—and it gives you a structured answer based on official standards. This MCP lets your agent act like a real-time consultant, handling everything from checking city metadata to auditing carrier names. By connecting this data through Vinkius, your AI client gets instant access to global aviation knowledge, letting you build robust travel and logistics applications with confidence.

Core Capabilities

01 — Auditing airport location details

Search for airports using their IATA code and retrieve key information like names and geographic coordinates.

02 — Verifying global airline codes

Confirm if an airline identifier is valid and retrieve its full name or operational metadata.

03 — Identifying aircraft types

Query the system to understand technical details about specific aircraft models used in global travel.

04 — Mapping city data by IATA code

Retrieve detailed geographical and administrative information for a city using its associated IATA code.

05 — Listing regional standards

Get a full list of all supported countries recognized within the global IATA catalog.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/iata-developer-portal** — connect your AI agent in three steps.

01 Subscribe to this MCP and input your unique IATA API Key into your AI client.

02 Instruct your agent on the specific aviation data you need—for example, 'List all supported countries' or 'Find details for LHR.'

03 Your agent executes the necessary tool call and returns structured metadata, giving you verifiable facts about airports, airlines, or aircraft.

The bottom line is that your AI client treats complex aviation databases like a simple conversational search engine.

Built For

This MCP is for developers and analysts who build tools around global travel, logistics, or industry standards. If you work with codes—airports, carriers, equipment types—and need data that isn't just guesswork, this is for you.

Travel Developer

Building booking engines or travel apps; they use the MCP to verify airline identifiers and audit aircraft types against official standards.

Logistics Planner

Designing supply chains that require knowing exact airport codes, coordinates, and country boundaries for operational routing.

Aviation Researcher

Performing rapid audits of global standards or historical data, needing the agent to query specific metadata like city details by IATA code.

What Changes When You Connect

01 Stop manually cross-referencing codes. Use `search_airports` to instantly audit global airports and get required metadata in one query.

02 Maintain data integrity by using `list_iata_countries` to ensure your application only references officially supported regional standards.

03 When building travel apps, verifying carrier legitimacy is key. The `search_airlines` tool ensures every airline identifier you use is valid.

04 Don't just get a name; get details. Use `get_city_details` with an IATA code to pull complete geographical and administrative information for planning.

05 If your app needs to track equipment, the `search_aircraft` function gives you technical metadata on specific plane types, streamlining logistics auditing.

Real-World Applications

Validating a multi-city itinerary

A travel developer needs to build an itinerary that crosses several international boundaries. They ask their agent to use `get_city_details` for every stop, ensuring the local city data is complete and accurate before committing to the booking API.

Building a logistics monitoring dashboard

An operations lead needs to confirm that all partner carriers are compliant. They ask their agent to use `search_airlines` repeatedly, quickly auditing every carrier ID against the master list to prevent shipping delays due to invalid codes.

Auditing a new market's regulatory scope

An aviation researcher needs to know which regions are supported by IATA. They ask the agent to run `list_iata_countries` first, giving them a comprehensive list of countries they must reference in their research.

Checking equipment compatibility for cargo routes

A planner needs to know what kind of planes are used on specific global routes. They ask the agent to use `search_aircraft` with a given IATA code, getting technical specs without consulting a manual.

Patterns to Avoid

Searching for airports via Google

✗ AVOID

Typing 'major international airport codes' into a general search engine gives you inconsistent results from various blogs or out-of-date guides. The data is messy and unreliable.

✓ INSTEAD

Use the `search_airports` tool with your agent. This guarantees that every code and piece of metadata comes directly from the official IATA standard.

Guessing airline identifiers

✗ AVOID

A developer might assume a local carrier's three-letter identifier is correct, but if it's outdated or unofficial, your booking system will fail. You can't afford guesswork.

✓ INSTEAD

Always run the `search_airlines` tool first. It verifies the code against the official record, giving you confidence in your data.

Using static reference lists

✗ AVOID

Relying on a local CSV file of countries means that if IATA adds or changes any region, your application becomes immediately outdated and potentially non-functional.

✓ INSTEAD

Use `list_iata_countries` to pull the most current list directly from the source. This keeps your data perpetually accurate.

The Right Fit

You should use this MCP if your core functionality relies on verifiable, globally standardized identifiers—airport codes, airline names, or country lists. If you need to confirm that a piece of input data matches an official industry standard (like checking `search_airports` for LHR), this is the right tool. Don't use it if you are just writing general text content or summarizing articles; those tasks require pure language models. Likewise, don't use it if your goal is to process unstructured documents—you need a document-parsing MCP instead. If your data source is proprietary (e.g., internal sales figures), this IATA portal won't help; you need a custom database connector.

Keeping Track of Global Aviation Codes Is a Nightmare

Right now, figuring out if an airport code is valid or what city it belongs to means jumping between three different databases. You copy the IATA code into one system for coordinates, then open another terminal tab to check the airline's full name, and finally cross-reference a third spreadsheet just to confirm the country.

With this MCP, your agent handles that entire sequence of checks in a single conversation. Instead of multiple clicks and messy copy-pasting, you ask one question—like 'What are the details for LHR?'—and get a clean, structured data package back instantly.

Getting Official Data with IATA Developer Portal MCP

The manual process of validating an entire fleet roster involves checking the plane model, then verifying its type code, and finally ensuring that type is compatible with the designated airport. Each step requires a different lookup and source.

Now, you use `search_aircraft` combined with `search_airports`. The agent pulls all required metadata—the aircraft specs matched to the location's rules—and presents it in one coherent answer. Your data is always grounded in official records.

IATA Developer Portal: 6 Tools

These tools let your agent access specific, audited functions for checking API status, finding city details, searching airlines, listing countries, querying aircraft, and locating airports.

#	TOOL	DESCRIPTION
01	`check_api_status`	Checks if the main IATA API is currently running and available for use.
02	`get_city_details`	Retrieves detailed geographical information about a city using its specific IATA code.
03	`search_airlines`	Looks up and verifies airline identifiers based on their unique IATA codes.
04	`list_iata_countries`	Generates a complete list of all countries that the IATA catalog currently supports.
05	`search_aircraft`	Provides technical metadata and details for specific aircraft types using their IATA codes.
06	`search_airports`	Searches for airports by their IATA code, returning key data like names and locations.

See It in Action

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

U Search for airport with IATA code 'LHR' (London Heathrow) using IATA Portal.

I've retrieved the details for London Heathrow Airport. It is located in the United Kingdom and uses timezone Europe/London. Would you like the full geographic coordinates or city metadata?

U Identify the airline with code 'BA' (British Airways).

I've identified the airline as British Airways. I can retrieve the full name and operational metadata for this carrier if you'd like.

U What aircraft corresponds to IATA code '744'?

I've retrieved the details. The IATA code '744' corresponds to the Boeing 747-400 (Passenger) aircraft. I can assist you with more aircraft metadata if needed.

Frequently Asked Questions

01 **What does search_airports do with IATA Developer Portal MCP?**

The `search_airports` tool finds airports based on their IATA code. It returns detailed metadata, including the airport's full name and its precise geographic location.

02 **Do I need an API key to use this MCP?**

Yes, you must subscribe to the portal and provide your unique IATA API Key in your client settings. This authorizes your agent to access the official data streams.

03 **Can I check if a country is supported using list_iata_countries?**

Yes, `list_iata_countries` gives you a comprehensive rundown of every country currently included in the IATA catalog. This helps keep your regional data scope controlled.

04 **How do I verify an airline using search_airlines?**

You pass the known or suspected IATA code to `search_airlines`. The tool verifies its existence and retrieves the carrier's official, full name for confirmation.

05 **Is this MCP better than a simple database lookup?**

This MCP is better because it allows your agent to orchestrate multiple lookups (e.g., city details + airport details) in one natural request, simulating a human expert's knowledge.

06 **What if I don’t know the IATA code for a city?**

You can use your agent to help structure the query. While `get_city_details` requires a code, the conversation itself allows you to ask for guidance on what codes are needed.

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 `"iata-developer-portal": { "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

IATA Developer Portal 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 IATA Developer Portal. 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	IATA Developer Portal MCP
Server ID	`019d8447-8832-71a2-b095-888df9139bfb`
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/iata-developer-portal.