MCP SERVER | NO CODE | CLOUD HOSTED

Travelport MCP

Book Flights and Hotels in a Single Conversation

Travelport MCP connects your AI agent directly to a global travel distribution system, allowing it to handle complex booking workflows for flights and hotels. Your agent can search real-time pricing, check hotel availability by location or ID, manage traveler details, validate payment cards, and commit entire reservations into official itineraries.

A+ Quality Score 100/100

flight-booking | hotel-reservations | travel-itinerary | gds | booking-management

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

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.

Travelport MCP

21 tools available
Cloud-hosted on Vinkius

This MCP lets your AI client perform the full lifecycle of travel planning, from initial idea to confirmed booking. You'll stop manually juggling multiple websites just to compare prices or check policies. Your agent can search for flights departing anywhere in the world, find available stays near specific landmarks, and build an entire trip itinerary right inside your chat window.

The process is smart: it doesn't just give you a list of options; it lets you create a reservation session—a 'workbench'—where you add travelers, select offers, and even validate payment cards before finalizing the booking. This means complex corporate travel or multi-city trips that used to take hours of clicking now happen in minutes. By hosting this connection within Vinkius, we make sure your AI agent has access to industry-leading capabilities, no matter which client you prefer.

Core Capabilities

01 — Search for flight options

Find available flights by performing specific searches or looking up the next segment of an existing itinerary.

02 — Check hotel availability and rules

Verify if a property has rooms open using location or ID, and retrieve detailed cancellation policies and house rules.

03 — Build and finalize reservations

Create a temporary booking session, add travelers and offers, and submit the final details to create a confirmed reservation (PNR).

04 — Validate payment details

Check if credit cards are valid or if billing addresses match the card's information before attempting a purchase.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Travelport credentials (Client ID, Secret, Username, Password, Access Group).

02 Use your AI client to initiate the process by requesting a new reservation session or starting a search for flights/hotels.

03 The agent collects all necessary data—like traveler names, dates, and payment info—and executes the required steps to return a confirmed itinerary or updated status.

The bottom line is you get an automated assistant that handles the entire travel booking process without needing to switch between separate industry platforms.

Built For

This MCP is essential for corporate travel managers and agents who spend all day navigating disparate systems. If you're tired of copying dates, searching three different vendor sites, and manually confirming payment details across multiple interfaces, this tool saves your workflow.

Travel Agent

Uses the MCP to automate the search and booking process for clients using natural language prompts instead of manual GDS entry.

Corporate Travel Manager

Quickly retrieves existing reservation details or checks real-time flight availability across multiple employees' trips without logging into separate systems.

Developer / Workflow Engineer

Integrates complex, multi-step travel booking capabilities directly into custom AI applications and internal agent workflows.

What Changes When You Connect

01 You don't manually check multiple rates. The MCP allows you to use `price_offer_full` or `price_offer_reference` to get the final, accurate cost for any air offer.

02 Managing reservations is simple. Instead of starting from scratch every time, your agent first uses `create_workbench`, then adds all necessary details using `add_traveler_to_workbench`.

03 You bypass the risk of incorrect payments. Before committing, you run `validate_card` and `verify_address` to ensure payment data is secure and accurate.

04 Booking policies are transparent. When searching for lodging, use `get_hotel_rules` so your agent can provide clients with cancellation details right away.

05 The entire process is contained. You search using `search_flights` or `search_stays_by_location`, build the itinerary, and finish everything by calling `commit_workbench`.

Real-World Applications

Booking a multi-city business trip

A corporate manager needs to book flights across four cities and three nights of hotel stays. They ask their agent, and the agent uses `search_flights` for all legs, then runs `search_stays_by_location` for accommodations. Finally, it creates a workbench and commits everything in one go.

Handling an emergency booking

A travel agent needs to find the next available flight after a cancellation. They ask their agent, which uses `search_next_leg` to find replacement options, then recommends pricing it using `price_offer_full`.

Building a custom booking tool

A developer wants an internal app that handles travel. They use the MCP's tools like `create_workbench`, `add_traveler_to_workbench`, and `commit_workbench` to build reliable, structured code.

Confirming payment details for a client

Before confirming any booking, the agent runs `validate_card` and `verify_address`. This prevents errors and ensures the payment data is legitimate before running `authorize_card`.

Patterns to Avoid

Treating it like a simple search API

✗ AVOID

The user only asks for flight options, gets them, and assumes the trip is booked because they saw prices. They forget the payment step.

✓ INSTEAD

Always remember to run `create_workbench` first, add all necessary offers (`add_offer_to_workbench`), and finally call `commit_workbench`. Never assume a search result means a confirmed booking.

Ignoring hotel policies

✗ AVOID

Booking a stay based only on the price without knowing if it's refundable or non-refundable, leading to client disputes.

✓ INSTEAD

Before committing any hotel reservation, always use `get_hotel_rules` so your agent can read out the specific cancellation and policy terms.

Assuming card validity

✗ AVOID

Attempting a booking with a credit card number that is expired or uses an old billing address.

✓ INSTEAD

Run `validate_card` and then use `verify_address` before initiating payment authorization. This prevents the whole process from failing at the last second.

The Right Fit

Use this MCP if your workflow requires more than just finding prices; you need to perform a full, structured booking cycle, including creating temporary reservation sessions and committing payments. If you only need simple data retrieval—like 'What is the price of this specific flight?'—then basic search tools might suffice. However, if that single task requires checking availability, validating payment details, adding travelers, and finally confirming the transaction to get a PNR, then this MCP is necessary. Do not use it just because you need to find flights; you must be prepared to build out the entire booking process from start to finish.

The hassle of coordinating travel details across multiple websites is exhausting.

Today, planning a complex trip means opening three different tabs: one for flights, one for hotels, and another for payments. You copy dates from the flight site into the hotel site, manually cross-reference policies, and then finally hop over to a separate payment portal. It’s slow, it's error-prone, and you spend half your time copying data.

With this MCP connection, that entire manual process collapses into one conversation. Your AI client handles the multi-step logic: finding flights using `search_flights`, checking hotels via `search_stays_by_location`, gathering traveler details, validating cards with `validate_card`, and confirming everything by calling `commit_workbench`. You get a confirmed result without leaving the chat.

Travelport MCP: Full Booking Control

Specific manual steps that vanish include copy-pasting reservation IDs, toggling between price quotes and actual bookings, and manually checking credit card expiration dates. Your agent does all of this logic in the background.

It's not just about finding deals anymore; it’s about execution. This MCP gives your AI client transactional capability—the ability to manage payments, create official records, and finalize multi-service trips reliably.

Travelport: 21 Tools for Global Booking

These tools give your AI client granular control over every step of the travel process—from initial searches to final payment authorization.

#	TOOL	DESCRIPTION
01	`add_offer_to_workbench`	Attaches a specific air flight offer to an existing reservation work session.
02	`add_traveler_to_workbench`	Adds the necessary details for one or more travelers to the current booking session.
03	`authorize_card`	Temporarily checks if a payment card is valid and can be used for authorization.
04	`check_hotel_availability`	Verifies which hotels have rooms open using the current version's search parameters.
05	`commit_workbench`	Finalizes all details in a temporary booking session and submits it to create a confirmed reservation.
06	`create_hotel_reservation`	Creates a new hotel booking using the current version's parameters.
07	`create_workbench`	Starts a brand-new, temporary reservation session to begin building an itinerary.
08	`get_hotel_rules`	Retrieves the specific rules and policies for a selected hotel stay.
09	`price_offer_full`	Calculates the total, final cost for an air flight offer using current product data.
10	`price_offer_reference`	Gets a price estimate for an air flight offer based on a reference catalog item.
11	`retrieve_flight_reservation`	Fetches the details and status of an existing, booked flight reservation.
12	`retrieve_hotel_reservation`	Retrieves the full details and current status of a confirmed hotel booking.
13	`reverse_authorization`	Voids a previous payment authorization, effectively canceling the hold on funds.
14	`search_flight_specific`	Performs a detailed search for flights using specific criteria inputs.

#	TOOL	DESCRIPTION
15	search_flights	Searches the entire catalog of products to find available flight options.
16	search_next_leg	Finds and prices the next segment or leg of a multi-day flight itinerary.
17	search_stays_complete	Performs a comprehensive search to find available hotel stays based on location and dates.
18	search_stays_by_id	Searches for specific hotel properties using their unique ID number.
19	search_stays_by_location	Finds available hotels by searching within a defined geographic area.
20	validate_card	Checks if a provided payment card number is syntactically valid and active.
21	verify_address	Confirms that the provided billing address matches the information recorded on the payment card (AAVS).

See It in Action

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

U Search for flights from JFK to LHR departing on October 15th and returning on October 22nd.

I've found several options for your trip. The best-priced offering is with British Airways at $650. Would you like me to price this specific offer or look for the next leg options?

U Find available hotels in Tokyo for a 3-night stay starting tomorrow.

Searching Tokyo stays... I found 12 available properties. The 'Park Hyatt Tokyo' has availability starting at $450/night. Shall I check the specific availability and rules for this hotel?

U Create a new booking workbench and add traveler John Doe to it.

Workbench created successfully (ID: WB-98765). I have added John Doe as the primary traveler. What flight or hotel offer should I add to this session?

Frequently Asked Questions

01 **How do I start a new booking with the Travelport MCP?**

You begin by calling `create_workbench`. This initializes a temporary session ID that allows your agent to build the itinerary step-by-step before making any commitments.

02 **Can I check if an existing reservation is still valid using Travelport MCP?**

Yes. You can use `retrieve_flight_reservation` or `retrieve_hotel_reservation` to pull up the current status and details of any booked trip.

03 **Does the Travelport MCP handle payments directly?**

It manages the payment flow. First, you use `validate_card` and then, if everything is good, you run `authorize_card` before committing to the final booking.

04 **What's the difference between searching for hotels by location vs. ID using Travelport MCP?**

Searching by location (`search_stays_by_location`) is useful when you only know the neighborhood, while `search_stays_by_id` lets you check a specific property even if you don't know its physical address.

05 **How do I get prices for an air offer using Travelport MCP?**

You have two options: use `price_offer_full` to calculate the total cost based on current products, or use `price_offer_reference` if you are basing the price on a catalog item.

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 `"travelport": { "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

Travelport 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 Travelport. 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	Travelport MCP
Server ID	`019e38fe-d635-7308-bc29-eb1c31752f70`
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/travelport.