MCP SERVER | NO CODE | CLOUD HOSTED

TomTom MCP

Calculate routes, check traffic, and map global locations.

TomTom equips your AI agent with global mapping, routing, and real-time traffic data. Convert any address to precise coordinates, calculate driving times between two points, or map live accidents within a specific area using this MCP.

A+ Quality Score 100/100

mapping | routing | geocoding | traffic-data | location-services | logistics

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

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.

TomTom MCP

10 tools available
Cloud-hosted on Vinkius

This connector lets you bring enterprise-grade location intelligence directly into your chat workflow. You can turn messy physical addresses into exact latitude/longitude pairs, and vice versa, so your agent always knows precisely where something is. Need to plan a trip? Input two coordinates, and the tool calculates the optimal route polyline, total distance, and estimated travel time, factoring in current conditions. It even maps out live traffic incidents—like accidents or construction zones—within any bounding box you define. If you're managing assets, you can figure out exactly how far your fleet can drive given a set time limit. Connecting this MCP through Vinkius gives your AI client immediate access to thousands of location services, making complex logistics questions simple chat queries.

Core Capabilities

01 — Convert Addresses and Coordinates

Translate any street address into accurate geographic coordinates or reverse the process to find a street name from given coordinates.

02 — Plan Optimal Routes

Calculate the necessary route, distance, and time needed for travel between two precise points using this MCP.

03 — Monitor Real-Time Traffic

Scan a defined area to identify live traffic incidents or assess the current speed and quality of a specific road segment.

04 — Find Local Points of Interest

Search for categories like hospitals or gas stations near a location, retrieving key contact details along the way.

05 — Determine Operational Boundaries

Calculate the maximum area (polygonal limit) your agents or vehicles can reach within a given time budget.

One Click on Vinkius — From Prompt to Execution

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

01 First, add this marketplace component to your workflow. Then, grab a secure API token from the TomTom Developer Portal.

02 Plug that key into the MCP and let your agent know what coordinates or addresses it needs to process.

03 Your AI client processes the request and returns structured data like polyline routes, traffic reports, or specific location metadata.

The bottom line is, you stop writing complex API calls and start asking natural questions about geography and logistics.

Built For

This MCP is critical for Logistics Engineers who need to model complex travel constraints or Fleet Managers who must continuously audit road conditions. If your job involves calculating anything based on physical location, you need this.

Logistics Engineer

You use the MCP to test route shedding and determine maximum operational time boundaries for new service areas.

Field Operations Manager

You ask your agent to check real-time traffic incidents across a region before dispatching routes, saving hours of manual checks.

App Developer

You programmatically pull local points of interest strings and their contact details directly into an application's database setup.

What Changes When You Connect

01 Stop guessing addresses. Use fuzzy_geocoding or structured_geocoding to guarantee precise coordinates from any input string, even if it's messy.

02 Get immediate route planning by calling calculate_route. You get the polyline geometry and travel time summary in one query, skipping multiple API calls.

03 Monitor road conditions instantly. Use get_traffic_incidents to map live accidents or use get_traffic_flow_segment to check if a specific avenue is backed up.

04 Never lose track of resources. The calculate_reachable_range tool tells you exactly how far your vehicles can go based on time and starting point.

05 Discover local services without leaving the chat. Search for POIs using search_poi_by_category, then get all the contact info via get_poi_details.

Real-World Applications

Dispatching a repair crew after an accident.

The manager asks: 'Check for traffic incidents near Lat 34.05, Lon -118.25.' The agent uses get_traffic_incidents to confirm an accident and then uses calculate_reachable_range to see if the crew can reach it within a two-hour window.

Mapping out service zones for expansion.

A developer asks: 'What area around this warehouse can we serve in 4 hours?' The agent calls calculate_reachable_range, providing the precise polygonal limit of potential new territory.

Validating customer-provided addresses.

The user inputs a handwritten address. The agent uses fuzzy_geocoding to convert it into validated coordinates, and then reverse_geocoding confirms the exact street name for billing records.

Optimizing delivery schedules in a new city.

The planner asks: 'What is the fastest route between these three points?' The agent uses calculate_route multiple times and compares results, factoring in current real-time traffic data for maximum efficiency.

Patterns to Avoid

Assuming coordinates are always perfect.

✗ AVOID

A user tries to calculate a route using only an address like 'Main Street' because they think the AI will figure it out.

✓ INSTEAD

First, use fuzzy_geocoding or structured_geocoding to convert that vague address into guaranteed latitude/longitude coordinates. Then, feed those precise numbers into calculate_route for accurate results.

Treating traffic data as simple on/off switches.

✗ AVOID

Asking if 'traffic is bad' without defining an area or time frame, leading to vague answers.

✓ INSTEAD

To get specific information, use a bounding box with get_traffic_incidents. If you need general flow quality, check the speed using get_traffic_flow_segment.

Trying to find POIs without knowing what type they are.

✗ AVOID

Simply asking 'what's around here?' results in an overwhelming list of irrelevant businesses.

✓ INSTEAD

Be specific. Use search_poi_by_category and name the category (e.g., 'pharmacy'). Then, use get_poi_details to pull out opening hours.

The Right Fit

Use this TomTom MCP if your workflow is fundamentally about physical location: routing, time-based service areas, or geo-validation. If you need to calculate how far a vehicle can go in 3 hours, use calculate_reachable_range. If you just need the straight line distance between two known coordinates, calculate_route works too. Don't use this if your problem is purely data aggregation (like joining spreadsheets) or complex natural language reasoning without location context; those require different tools. However, if you are dealing with vague inputs and need to ensure coordinate accuracy before any calculation, structured_geocoding is your required first step. If all you have are coordinates but need a street name for billing, reverse_geocoding handles that perfectly.

Mapping out logistics without constantly switching between tools.

Right now, planning any complex multi-stop route means juggling five different tabs: one for the map, one for traffic alerts, one for address verification, and three others just to copy/paste coordinates. You spend more time managing the data flow than actually solving the problem.

With this MCP, you simply ask your agent a question like 'What's the best way to get from Point A to Point B right now?' The connector handles the geocoding, the routing calculation, and checks for real-time traffic conditions. You get a single, actionable answer.

TomTom MCP delivers immediate operational context.

You no longer have to manually verify addresses using fuzzy_geocoding before running calculate_route, nor do you need separate searches for POIs and their contact details. All that data is available in one sequence of operations.

The difference is the complexity level. You move from writing multi-step code blocks with external API calls to asking a single, intelligent question.

TomTom: 10 Location Services Tools

Use these tools to calculate routes, estimate travel times, check live traffic flow, or find points of interest anywhere in the world.

#	TOOL	DESCRIPTION
01	`calculate_reachable_range`	Determines the full area that can be covered, given a starting point and maximum travel time in seconds.
02	`calculate_route`	Calculates the direct route polyline, total distance, and estimated driving time between two specific coordinates.
03	`search_poi_by_category`	Locates points of interest (POIs) near a coordinate by specifying a category like 'hospitals' or 'fuel'.
04	`fuzzy_geocoding`	Converts an address string into precise coordinates, even if the input address has minor spelling errors.
05	`get_poi_details`	Retrieves detailed information and rich metadata for a specific point of interest ID found earlier.
06	`get_traffic_incidents`	Pulls real-time details on traffic incidents, like accidents or construction, within any defined map boundaries.
07	`get_traffic_flow_segment`	Retrieves the current speed and overall quality of travel for a specific segment of road at a given coordinate.
08	`reverse_geocoding`	Converts geographic coordinates back into a readable, formatted street address string.
09	`autocomplete_place_search`	Offers predictive location suggestions as you type a partial place name or address segment.
10	`structured_geocoding`	Converts an address into coordinates using required components like the country code and postal zip code for accuracy.

See It in Action

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

U Convert these coordinates into an address: Lat 40.7128, Lon -74.0060.

The coordinates return an address explicitly linked to the parameter grid:
- Broad St, New York, NY 10004, United States

U Check for any traffic incidents on the 101 freeway bounded roughly by these dimensions.

A real-time scan revealed 1 active traffic incident in that bounding box:
- Type: Accident/Lane blocked
- Severity: Major delay expected
- Center: Lat 34.05, Lon -118.25

Frequently Asked Questions

01 **How does TomTom MCP handle bad addresses?**

It uses fuzzy_geocoding and structured_geocoding. These tools can convert physical address strings into accurate coordinates even if the input is messy or incomplete.

02 **Can I find traffic information for a whole region?**

Yes, you use get_traffic_incidents by providing minimum and maximum latitude/longitude boundaries. This gives you an overview of accidents and jams in that entire box.

03 **Is calculate_route just straight-line distance?**

No. calculate_route calculates the actual polyline geometry, total distance, and estimated travel time based on typical road networks, not just a straight line through space.

04 **How do I get contact information for a business near me?**

First, use search_poi_by_category to locate the point of interest. Then, call get_poi_details using its unique ID to pull out rich metadata like phone numbers and hours.

05 **Does TomTom MCP account for travel time limitations?**

Yes. The calculate_reachable_range tool determines the exact polygonal area you can reach given a specific time budget in seconds.

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 `"tomtom": { "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

TomTom 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 TomTom. 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	TomTom MCP
Server ID	`019d7614-3803-7163-a858-cae0c9e01304`
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/tomtom.