MCP SERVER | NO CODE | CLOUD HOSTED

FullStory MCP

Analyze deep digital behavior through conversation.

FullStory FullStory MCP lets your AI agent analyze deep user behavior and digital experience intelligence. You can ask it to fetch specific session playbacks, audit user profiles across devices, or track custom events directly from chat. It turns complex analytics tasks—like finding why a checkout flow failed—into simple questions for your agent.

A+ Quality Score 98.33/100

digital-experience | session-replay | product-analytics | user-behavior | event-tracking | session-metadata

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

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.

FullStory MCP

11 tools available
Cloud-hosted on Vinkius

Figuring out where users drop off or what features they ignore used to mean jumping between dashboards and manually piecing together timelines. Now, you can connect your FullStory account directly through this MCP. Your AI client handles the heavy lifting of product analytics, letting you query complex data—like tracking a specific click or pulling up a user's full activity history—using natural conversation. It gives you immediate visibility into how people use your site or app.

Whether you need to check a customer's profile across multiple devices or programmatically log an event that bypassed the front-end code, this MCP handles it all. When you subscribe via Vinkius, your agent gets instant access to FullStory's entire data set of user interactions and session records. You stop building complex custom scripts just to answer basic questions about user journeys.

Core Capabilities

01 — Audit User Profiles

Get a full profile history for any specific user, including their total duration on the site and all associated custom variables.

02 — Search & Retrieve Sessions

List or search multiple session recordings based on criteria like date range or browser type, and retrieve deep metadata for a single recording.

03 — Analyze Interaction Events

Fetch the entire chronological list of actions—clicks, navigation changes, and custom events—that happened during any recorded session.

04 — Programmatic Data Logging

Inject custom backend metrics or automatically update user identities into FullStory using simple commands.

05 — Manage Compliance & Exports

Delete specific user data to meet privacy regulations like GDPR, or list raw data bundles ready for deep analytical processing.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP through the Vinkius Marketplace and enter your FullStory API Key.

02 Tell your AI agent what you need, like 'Show me all events for user 123.'

03 The agent executes the request against FullStory and returns the structured data right back in the chat.

The bottom line is that your AI client talks directly to FullStory's core data layer without needing manual API calls or complex integrations.

Built For

This MCP is essential for Product Managers and support teams who are tired of manually cross-referencing dashboards just to answer a basic question about user behavior. If you regularly need context on *why* a customer failed at checkout, this tool saves hours of clicking.

Product Manager

You use it to quickly check activity or find specific session playbacks to validate hypotheses about product usage patterns.

Support Engineer

You get a real-time, detailed overview of a customer's recent interaction events while troubleshooting an issue and speaking directly with the user.

Data Analyst

You use it to pull structured data points—like listing segments or checking full profile metadata—to feed into secondary analysis models.

What Changes When You Connect

01 Pinpoint user failure points. Instead of guessing, you can use the agent to fetch the complete set of captured events for a session ID, showing exactly where clicks or errors occurred.

02 Maintain data integrity easily. You don't have to manually update records; just ask your agent to run `create_update_user` to sync custom tenant data into FullStory tracking.

03 Understand the customer journey at scale. Use `list_sessions` and `get_session` to search for multiple recordings, providing playback links and metadata overlays without touching a dashboard.

04 Ensure compliance automatically. If you need to delete sensitive data quickly, running `delete_user` is one simple command that handles the removal of telemetry and session history.

05 Improve data quality from your backend. You can now programmatically log custom interactions using `create_event`, enriching FullStory datasets with server-side metrics.

Real-World Applications

A critical checkout flow failed for a premium user.

The support engineer asks the agent: 'What happened during the last 10 minutes of activity for this customer?' The agent uses `get_session_events` and provides a detailed list showing the exact navigation mutation that triggered the error, allowing immediate resolution.

We suspect our new sign-up flow is confusing users.

The product manager asks the agent to 'List all users who started but didn't finish onboarding.' The agent uses `list_users` and then pulls detailed profiles using `get_user` for a small sample group, showing custom properties like total spent.

We need to prove GDPR compliance for historical data.

The privacy officer simply instructs the agent: 'Delete all records associated with user Jane Doe.' The agent executes `delete_user`, logging the action and confirming the permanent removal of sensitive telemetry.

We launched a new feature that needs tracking from our API.

Instead of waiting for front-end changes, the developer asks the agent to 'Track an event called Payment Processed.' The agent runs `create_event`, ensuring the backend data is logged correctly into FullStory immediately.

Patterns to Avoid

Trying to analyze raw logs manually

✗ AVOID

A team member tries to find out why a user exited by searching through multiple raw API endpoints and piecing together timestamps from different services.

✓ INSTEAD

Just ask your agent. For instance, tell it to `list_sessions` for the last week, then select a session ID, and finally run `get_session_events` on that specific recording.

Forgetting to update identity records

✗ AVOID

The system sees 'User A' logging in from one device but fails to link subsequent actions because the profile was never updated with a new ID.

✓ INSTEAD

Before analyzing, run `create_update_user` to synchronize any custom tenant data. This ensures all sessions are mapped under the correct user identity.

Assuming frontend coverage for server actions

✗ AVOID

A backend process completes successfully but has no record of that activity in FullStory's timeline, leading to inaccurate analytics.

✓ INSTEAD

Use `create_event` to programmatically inject the necessary metrics. This guarantees the data appears in the user timeline even if front-end JavaScript can't capture it.

The Right Fit

Use this MCP if your job requires answering specific, deep questions about *user behavior* (e.g., 'What did they click?' or 'Why did they fail at checkout?'). It excels when you need to pull granular data like `get_session_events` or manage compliance via `delete_user`. Don't use it if your goal is general dashboard reporting, which should stay in the main FullStory UI. If you simply need a high-level count of users without context, listing segments (`list_segments`) works, but for deep troubleshooting, this MCP is unmatched because it lets your agent handle the complex data retrieval flow across multiple tools.

Manually connecting user actions to business problems takes forever.

Right now, figuring out why a feature isn't working involves clicking into the dashboard, filtering by date range, manually selecting different users, and then cross-referencing multiple views just to find one specific error or failed step. It’s slow, tedious work that requires dozens of clicks and copy-pasting.

With this MCP, you simply ask your agent a question—like 'Show me the full activity history for user X'—and it executes the necessary tools like `get_user` and `list_sessions`. You get the structured data instantly in plain text. No manual clicks, no dashboard hopping.

Get complete visibility into a customer’s journey with FullStory.

The process of auditing user activity used to require navigating through multiple tabs and joining datasets manually. You had to fetch the profile data, then separately pull the session metadata, and finally correlate those timestamps yourself.

Now you can ask for a comprehensive overview, letting your agent handle the complexity behind the scenes. It combines `get_user` with `list_sessions` in one conversational turn. The difference is immediate: contextually rich answers instead of isolated data points.

FullStory MCP: 11 Tools for Analytics

These tools let your AI agent perform specific actions in FullStory. Use them to manage user identity, retrieve sessions, or inject custom events into your analytics data.

#	TOOL	DESCRIPTION
01	delete_user	Permanently removes a user's private telemetry, property links, and session history from FullStory.
02	get_account_info	Validates the connection by retrieving basic platform statistics and defining the workspace scope.
03	get_session_events	Fetches a complete timeline of all captured events, including clicks, navigation changes, and DOM structure alterations for one session.
04	get_session	Retrieves core metadata about an entire recording, like total active time, page transitions, and event markers.
05	get_user	Gathers a user's full tracking profile, including custom variables and cross-device interaction boundaries.
06	list_exports	Lists the status of bulk data export jobs, showing processing states and download URLs for warehouse ingestion.
07	list_segments	Retrieves definitions of saved audience groups, including how many members are in those segments over the last 30 days.
08	list_sessions	Searches for session recordings by specific criteria and provides links to view them based on their telemetry data.
09	list_users	Queries the directory to list user identities, custom parameters, and aggregate behavioral profiles across your site.
10	create_event	Sends a custom backend event into FullStory that wasn't captured by the front-end JavaScript client.
11	create_update_user	Synchronizes or updates specific user identity data into FullStory to improve context for session tracking.

See It in Action

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

U List the last 5 sessions for user UID 'user_12345'.

Retrieving sessions... I found 5 recent recording sessions for UID user_12345. Key sessions include a 'Checkout' flow from yesterday and a 'Login' event from this morning. Which playback link would you like?

U Show me the full profile details for FullStory ID '123:456'.

Fetching user data... For ID 123:456, I found email 'customer@example.com'. They are part of the 'Premium' segment and have 15 custom properties, including 'total_spent: 150.00'.

U Track a server event 'Subscription Renewed' for user 'johndoe@email.com'.

Event tracked! I have successfully posted the 'Subscription Renewed' event to FullStory for johndoe@email.com. This data will now appear in their user timeline.

Frequently Asked Questions

01 **How do I get an API Key for FullStory?**

Log in to FullStory, navigate to Settings > API Keys, and click on 'Create Key' to generate a new token with Admin or Architect permissions.

02 **What is the difference between ID and UID?**

The 'id' is a system-generated unique identifier from FullStory, while the 'uid' is your application's own internal user ID passed via FS.identify.

03 Can I watch session playbacks through the agent?

The agent can retrieve the 'fsUrl' for any session, which is a direct link to watch the playback in your browser. You cannot watch the video inside the chat interface directly.

04 Are custom user properties supported?

Yes! When you use 'upsert_user' or 'get_user_details', all custom properties stored in the 'properties' object are accessible and manageable via the agent.

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 `"fullstory": { "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

FullStory 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 FullStory. 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	FullStory MCP
Server ID	`019d75a1-c088-7070-99cb-bc8ad93d6476`
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/fullstory.