MCP SERVER | NO CODE | CLOUD HOSTED

Apple App Store MCP for AI Agents

Manage app data, versions, and sales reports in App Store Connect

Apple App Store MCP connects your AI agent directly to App Store Connect data. Monitor app health, track customer reviews across all territories, and analyze sales trends for your published iOS apps without leaving your workflow.

A+ Quality Score 100/100

app-store-connect | ios-development | testflight | app-metadata | sales-reporting | customer-reviews

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/apple-app-store

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.

Apple App Store MCP

8 tools available
Cloud-hosted on Vinkius

Managing an app's lifecycle means juggling builds, tracking feedback, and monitoring revenue—all inside a complex portal. This MCP gives your AI client direct access to that data, making it actionable via natural language commands. You can monitor everything from TestFlight build statuses to aggregated sales reports with simple prompts. The goal is to take the manual heavy lifting out of app management. Instead of clicking through multiple tabs and downloading CSVs, you talk to your agent and get a synthesized answer right away. When you connect this MCP via Vinkius, your AI client treats App Store Connect like another API endpoint, letting you treat all your app data—metadata, versions, reviews, and sales figures—as one single source of truth.

Core Capabilities

01 — Manage App Inventory

List every app in your account and pull detailed metadata for any specific product.

02 — Analyze Customer Feedback

Access and summarize customer reviews, including star ratings, across all global territories.

03 — Track App Versions

View the complete history of your app's submissions, release notes, and current status in App Store Connect.

04 — Monitor TestFlight Builds

Check the status of pre-release versions and monitor ongoing beta testing builds for specific apps.

05 — Review Sales Performance

Download aggregated sales reports detailing download counts and revenue performance over daily, weekly, monthly, or yearly intervals.

06 — Get Specific App Details

Retrieve key information for a single app title, including its current metadata and identifiers.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/apple-app-store** — connect your AI agent in three steps.

01 Connect your preferred AI client to the Apple App Store MCP via Vinkius.

02 Prompt your agent with a natural language request, like 'Show me the average rating for my latest build.'

03 The MCP executes the necessary query against App Store Connect and returns the data structure to your agent.

The bottom line is that your AI client handles all the API calls; you just ask it questions about your app's performance, and it gives you a direct answer.

Built For

This MCP is built for people whose job revolves around publishing apps. If you spend hours digging through App Store Connect to compile status reports or figure out why sales dipped last week, this is for you.

iOS Developer

Uses the MCP to check build statuses and pull version history before a major release.

Product Manager

Gathers user feedback (list_customer_reviews) and cross-references it with app metadata to plan V2 features.

Marketing Director

Monitors sales trends (get_sales_reports) and checks App Store visibility metrics to optimize store presence.

What Changes When You Connect

01 Stop manually checking build statuses. Use list_prerelease_versions to instantly know if your TestFlight builds are ready for testing.

02 Get immediate feedback on public sentiment. Running the list_customer_reviews tool allows you to analyze what users love—or hate—right now.

03 Understand revenue performance at a glance. The get_sales_reports function lets you track downloads and money made over any time period (daily, monthly, yearly).

04 Keep your records clean. Use get_app_store_versions to pull the full history of release notes and submission statuses without digging through archives.

05 Know what you're working with. Running list_apps gives you a complete inventory check across all products in your account.

Real-World Applications

Investigating a Sales Dip

A marketing director noticed revenue dropped last week. Instead of emailing the finance team, they ask their agent to run get_sales_reports for the previous period. The agent returns structured data showing the drop correlated with a specific geographical region.

Preparing for V2 Launch

A product manager needs to justify new features. They ask the agent to compare list_customer_reviews from the last quarter against get_app_info data, identifying common feature requests mentioned by users.

Debugging a Release Issue

An iOS developer finds that their latest build isn't showing up. They use list_builds and then list_prerelease_versions to confirm the exact build ID and status, letting them fix the problem faster.

Auditing App Metadata

A marketing team needs to check if all their apps are using the right keywords. They run get_app_info for multiple titles to ensure consistency across the entire catalog.

Patterns to Avoid

Confusing App Store Connect with Google Play

✗ AVOID

Trying to use a generalized app management tool that doesn't understand Apple's unique versioning or review structure, leading to missing data points.

✓ INSTEAD

Use the dedicated list_customer_reviews and get_app_store_versions tools. They are built specifically for App Store Connect rules, ensuring you only see accurate iOS-specific data.

Overlooking Build Stages

✗ AVOID

Assuming that a build is live when it's actually stuck in review or needs manual testing before release.

✓ INSTEAD

Always confirm status using list_builds and list_prerelease_versions. This confirms the exact stage, whether 'Ready for Testing' or 'In Review'.

Treating Sales Data as Static

✗ AVOID

Running a one-time report without specifying the time frequency (DAILY vs. YEARLY), resulting in an unreadable data dump.

✓ INSTEAD

Always specify the reporting cadence when using get_sales_reports. You can ask for 'weekly reports' or 'monthly trends' to keep the output clean.

The Right Fit

You should use this MCP if your workflow requires querying specific, structured data from App Store Connect—like historical sales numbers (get_sales_reports) or comparing user sentiment across multiple app versions. It's perfect when you need to synthesize 'Why did sales drop?' by cross-referencing review data with revenue data. Don't use this if all you want is a high-level summary of your product roadmap; that's for general project management tools. If you just need to track simple task completion, an internal ticketing system might suffice. But if the data lives in Apple's ecosystem, this MCP is non-negotiable.

App Store Connect Data Analysis with Apple App Store MCP

Today, checking your app’s performance requires a painful process: logging into App Store Connect, opening the sales tab for revenue trends, then switching to the customer reviews section to read feedback. You spend time copy-pasting data points—downloading a weekly report and manually summarizing the star rating changes.

With this MCP, you just ask your agent to compare last month's download numbers with sentiment from new reviews. It pulls the sales figures using get_sales_reports and combines them with list_customer_reviews results into one clean summary. You get immediate insights instead of a dozen tabs open.

Managing App Versions via Apple App Store MCP

Before, tracking versions meant navigating the 'App Store' section to check if your release notes were correct and if the build was actually available for beta testing. It was a guessing game of dates and status messages.

Now, you ask about version history or pre-release builds. The MCP runs list_app_store_versions or list_prerelease_versions, providing structured details on exactly what was submitted, when it happened, and if it's ready for the next round of testing. It’s all visible instantly.

Apple App Store Data: 8 Tools for Publishing Metrics

These tools allow your agent to perform specific actions like checking account status, listing all apps, tracking builds, or downloading sales reports.

#	TOOL	DESCRIPTION
01	`get_account_check`	Verifies that the connection to your App Store Connect account is active and authorized.
02	`list_apps`	Retrieves a list of all published apps associated with your account ID.
03	`list_builds`	Lists all compiled builds that exist for a specific app title, tracking their status and version.
04	`list_customer_reviews`	Pulls customer reviews for a given app ID so you can analyze public sentiment.
05	`get_sales_reports`	Downloads comprehensive reports on sales and download trends, allowing you to specify the time frequency (DAILY, WEEKLY, MONTHLY, YEARLY).
06	`get_app_info`	Retrieves detailed metadata, including keywords and descriptions, for one specific app.
07	`get_app_store_versions`	Lists all historical App Store versions associated with an app, detailing release notes and submission status.
08	`list_prerelease_versions`	Displays the current list of pre-release builds for beta testing using TestFlight.

See It in Action

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

U What were my top 3 performing apps last month?

App Performance Summary (May 2024)

APP NAME	DOWNLOADS	REVENUE TREND
Fitness Tracker	15,200	$→$ Up 8%
Recipe Master	9,100	$↗$ Stable
Expense Guide	6,400	$↓$ Down 5%

The main driver for the drop in 'Expense Guide' was a spike in negative reviews regarding the new UI layout.

U List all my current apps and their latest available build status.

App Inventory & Build Status

You currently manage 4 titles:
Fitness Tracker: *Latest version submitted (2.1.0). Status:* Ready for Testing*. Build ID: B-45.
Recipe Master: *Metadata available. Last build was 1.9.0, status:* Archived*.
Expense Guide: *Available builds: Beta v3.0. Status:* In Review*.
• **New Utility Tool:** No builds found yet.

U Show me the average rating and key complaints for 'Recipe Master'.

I pulled the 50 most recent reviews for Recipe Master.

⭐ **Average Rating: 4.3/5 Stars**

- **Praise Points:** Users consistently love the search filter functionality.
- **Key Complaints:** The primary issue is stability on older iOS versions, with multiple users mentioning crashes after updating.

Frequently Asked Questions

01 **How do I use the Apple App Store MCP to check my app's sales data?**

You can get detailed revenue and download reports by asking your agent to run a report. Just tell it what time frame you need, like 'weekly reports for Q2,' and it handles the complex reporting logic.

02 **Can this MCP track beta builds in TestFlight?**

Yes. It provides real-time visibility into your pre-release versions through list_prerelease_versions, letting you know if a build is 'Ready for Testing' or stuck in review.

03 **Does the Apple App Store MCP help me analyze customer reviews?**

Absolutely. It pulls recent customer feedback and allows your agent to summarize common themes, like specific features users praise or bugs they keep reporting.

04 **How do I find out all the apps I have published?**

You simply ask it to list all your apps. The MCP runs list_apps and gives you a complete inventory of every product connected to that App Store Connect account.

05 **Is this Apple App Store MCP good for developers working on V2?**

It's perfect. You can pull the current version history (get_app_store_versions) and then cross-reference user feedback with that history to prove why your new features are necessary.

06 **Does the Apple App Store MCP support different time periods for reports?**

Yes, it's flexible. When requesting sales data (get_sales_reports), you can specify if you want daily, weekly, monthly, or yearly aggregated statistics.

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 `"apple-app-store": { "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

Apple App Store 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 Apple App Store. 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	Apple App Store MCP
Server ID	`019d7550-4994-70e8-8a98-5b0e78c57d2d`
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/apple-app-store.