MCP SERVER | NO CODE | CLOUD HOSTED

OFX Bank Statement Parser MCP

Turn raw bank exports into secure, structured JSON.

The OFX Bank Statement Parser takes raw, archaic bank export files (OFX/QFX) and converts them into clean, structured JSON data. It's a secure financial bridge designed to let your AI client analyze complex transaction histories without you having to upload sensitive records to the cloud or wrestle with confusing SGML code. Your agent gets organized numbers, not raw text.

F Quality Score 3.6/100

financial-data | data-parsing | bank-statements | local-processing | data-extraction | transaction-history

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/ofx-bank-statement-parser

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.

OFX Bank Statement Parser MCP

1 tools available
Cloud-hosted on Vinkius

Dealing with bank statements is usually tedious enough without needing an LLM that can't read the format. Most public AI models fail when confronted with OFX or QFX files because they use an old-school structure (SGML) that confuses standard reading algorithms. This MCP solves that problem by acting as a local financial bridge. It parses your bank export file completely on your machine, extracting only clean transactional data—Date, Amount, Description, and Type—into a structured JSON array. Your AI agent never sees the messy raw file; it only gets the organized numbers it needs to do real work. This means you can ask complex questions like 'What was my spending pattern on travel last quarter?' and get an answer without compromising your financial data. By connecting this MCP through Vinkius, you keep all your sensitive information local while giving your AI client the best possible input for accounting or budgeting.

Core Capabilities

01 — Structure Raw Bank Files

It takes messy OFX/QFX files and converts them into clean JSON data that any program can easily read.

02 — Maintain Local Privacy

The parsing happens entirely on your machine, meaning zero cloud uploads of sensitive bank information.

03 — Extract Core Transaction Data

It isolates the key pieces of financial data—date, amount, description, and transaction type—from the noise.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/ofx-bank-statement-parser** — connect your AI agent in three steps.

01 You provide the absolute file path to your OFX or QFX bank statement file.

02 This MCP runs locally, parsing the complex structure and safely extracting only clean transactional data into a structured JSON array.

03 Your AI client receives the organized JSON data, allowing it to perform analysis without ever seeing the original raw file.

The bottom line is your agent gets perfectly formatted numbers ready for immediate analysis, keeping your bank details private all the way through.

Built For

Anyone whose job involves reviewing financial records needs this. Think accountants struggling with non-standard file formats or personal finance managers tired of manual data entry. If you spend time reconciling statements, this MCP saves hours of frustration.

Bookkeeper

They use this to process batches of bank exports, ensuring every transaction record is clean and structured for ledger entry.

Financial Analyst

They connect this MCP to run complex historical queries across multiple statements to identify spending trends or revenue gaps.

What Changes When You Connect

01 Stop worrying about data leaks. Since the parsing happens locally on your machine, you never have to upload sensitive bank statements to a public cloud service or agent.

02 Your AI client gets perfect input every time. It doesn't waste compute cycles guessing where one transaction ends and the next begins because the data is already structured JSON.

03 Handle any global bank export. This MCP supports all standard OFX or QFX formats, meaning it works with practically any financial institution you use.

04 Instantly answer complex questions. Instead of manually categorizing transactions in a spreadsheet, ask your agent: 'How much did I spend on dining last quarter?'

05 Streamline reconciliation. By getting clean transaction data—Date, Amount, Description, Type—you speed up the matching process between bank records and internal ledgers.

Real-World Applications

Analyzing a year of spending habits

A freelance accountant needs to analyze 12 months of client expenses. Instead of manually downloading, cleaning, and uploading dozen of statements, they provide the files via this MCP and ask their agent: 'Create a pivot table showing all expense categories for Q3.' The result is an immediate markdown table without any data handling risk.

Reconciling corporate credit cards

A finance manager needs to match raw bank exports against internal accounting software records. They run the transactions through this MCP, which spits out clean JSON, allowing their agent to easily cross-reference amounts and dates for reconciliation.

Calculating savings rates

A personal finance user wants to know if they are meeting a savings goal. They point the tool at their bank export and prompt: 'Calculate my total income versus expenses, then determine my net saving rate for this period.' The answer is delivered instantly.

Auditing historical transactions

An auditor needs to check all payments made to a specific vendor over several years. They use the MCP to extract data from multiple statements and ask their agent: 'List every transaction involving Vendor X, totaling the amount paid.' The clean JSON makes this query straightforward.

Patterns to Avoid

Pasting raw text into an AI chat

✗ AVOID

A user copies and pastes a section of their bank statement directly into the prompt box. The LLM struggles because it treats the file like unstructured text, misinterpreting field breaks or SGML tags.

✓ INSTEAD

You must use the `parse_ofx_bank_statement` tool. By providing the actual file path to this MCP first, you guarantee the data is clean JSON before your agent even sees it.

Using generic PDF parsers

✗ AVOID

A user tries to upload a bank statement saved as a screenshot or an image-based PDF. The AI fails because it has no way of knowing which numbers belong together.

✓ INSTEAD

This MCP requires the original OFX/QFX file format, ensuring the data integrity is maintained from the source export.

Attempting cloud processing

✗ AVOID

A user connects their bank account directly to an AI agent for analysis. This exposes highly sensitive credentials and financial details to a third-party server.

✓ INSTEAD

This MCP processes the file locally on your machine, keeping all raw data entirely air-gapped from the cloud.

The Right Fit

Use this MCP if you have bank statements in standard OFX or QFX format and need to perform deep analysis while maintaining absolute privacy. It's perfect for accounting professionals who deal with unstructured, high-volume financial exports.

Don't use it if your data is stored as photographs, screenshots, or general PDF reports (you'll need a different type of OCR tool). Also, do not use this MCP just to summarize an email chain; its sole purpose is structured transaction extraction. If you only need basic text reading and don't care about the underlying financial format, any generic file reader will suffice. But if data structure and privacy are non-negotiable, this tool is what you need.

The Pain of Manual Financial Data Preparation

Today, getting clean transactional data means opening multiple exports, manually copying amounts into a spreadsheet, and then trying to teach your AI agent the specific quirks of each bank's file structure. You spend time cleaning up formatting errors instead of analyzing actual spending patterns.

With this MCP, you feed the raw export file into one secure process. The system handles all the messy parsing, leaving only perfectly structured JSON data for your agent to read. You get clean, ready-to-analyze numbers in seconds.

Getting Clean Transaction Data with parse_ofx_bank_statement

The steps that vanish are the constant copy-pasting of dates and amounts, the cross-checking of headers, and the guesswork involved in reading confusing SGML tags. The whole process shrinks down to one single prompt.

You stop dealing with messy files and start talking directly to actionable insights. It's a massive difference: you go from data preparation hell to pure analysis.

OFX Bank Statement Parser: 1 Tool Available

The listed tool allows you to take raw OFX or QFX files and reliably extract all necessary transaction details into clean, organized JSON data.

#	TOOL	DESCRIPTION
01	`parse_ofx_bank_statement`	This tool takes a specified OFX or QFX file path and outputs clean, structured JSON containing all transactions.

See It in Action

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

U Read my statement.ofx and categorize all my expenses into a markdown table.

Here is your categorization:

CATEGORY	TOTAL SPENT
Food & Dining	$450.00
Transport	$120.00

U Look at my bank export and find out exactly how much I paid to 'AWS' last year.

I found 12 transactions matching 'AWS' in your export, totaling exactly $1,450.32.

U Analyze my monthly income versus expenses and calculate my savings rate.

Based on the OFX data, your total income was $5,000 and total expenses were $4,000. Your savings rate for this period is 20%.

Frequently Asked Questions

01 **Does the OFX Bank Statement Parser handle QFX files?**

Yes, it supports both standard OFX and QFX formats. This MCP is designed to read multiple types of common bank exports so you don't have to worry about which format your bank uses.

02 **Is the data processed locally when using parse_ofx_bank_statement?**

Yes, that's a core feature. The parsing happens entirely on your machine; your sensitive financial details never leave your local environment and are not uploaded to any cloud service.

03 **What kind of data does the JSON output contain?**

The resulting JSON array contains clean, structured fields for every transaction: Date, Amount, Description, and Transaction Type. This is exactly what your agent needs to run calculations.

04 **Can I use this MCP with my personal bank statements?**

Absolutely. Because the process is local, it's perfect for personal finance management. You can feed it any OFX file from a major global bank and get structured data back.

05 **Does parse_ofx_bank_statement require me to write code?**

No, you simply provide the file path through your AI client. The MCP handles all the complex parsing logic in the background so you just get clean data ready for a prompt.

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 `"ofx-bank-statement-parser": { "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

OFX Bank Statement Parser 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 OFX Bank Statement Parser. 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	OFX Bank Statement Parser MCP
Server ID	`019e38cb-44ad-739e-b5f9-9779d6a9c907`
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/ofx-bank-statement-parser.