Open in app ↗

Medium

Generative AI · Follow publication

Member-only story

I Built a Wall Street Analyst in 200 Lines of Code — And It Outperformed My $2,000/Month Bloomberg Terminal

How an open-source AI agent named Dexter is democratizing financial research, one autonomous query at a time

7 min read · 4 days ago

Adham Khaled Following

Listen Share More

The cursor blinked mockingly at me.

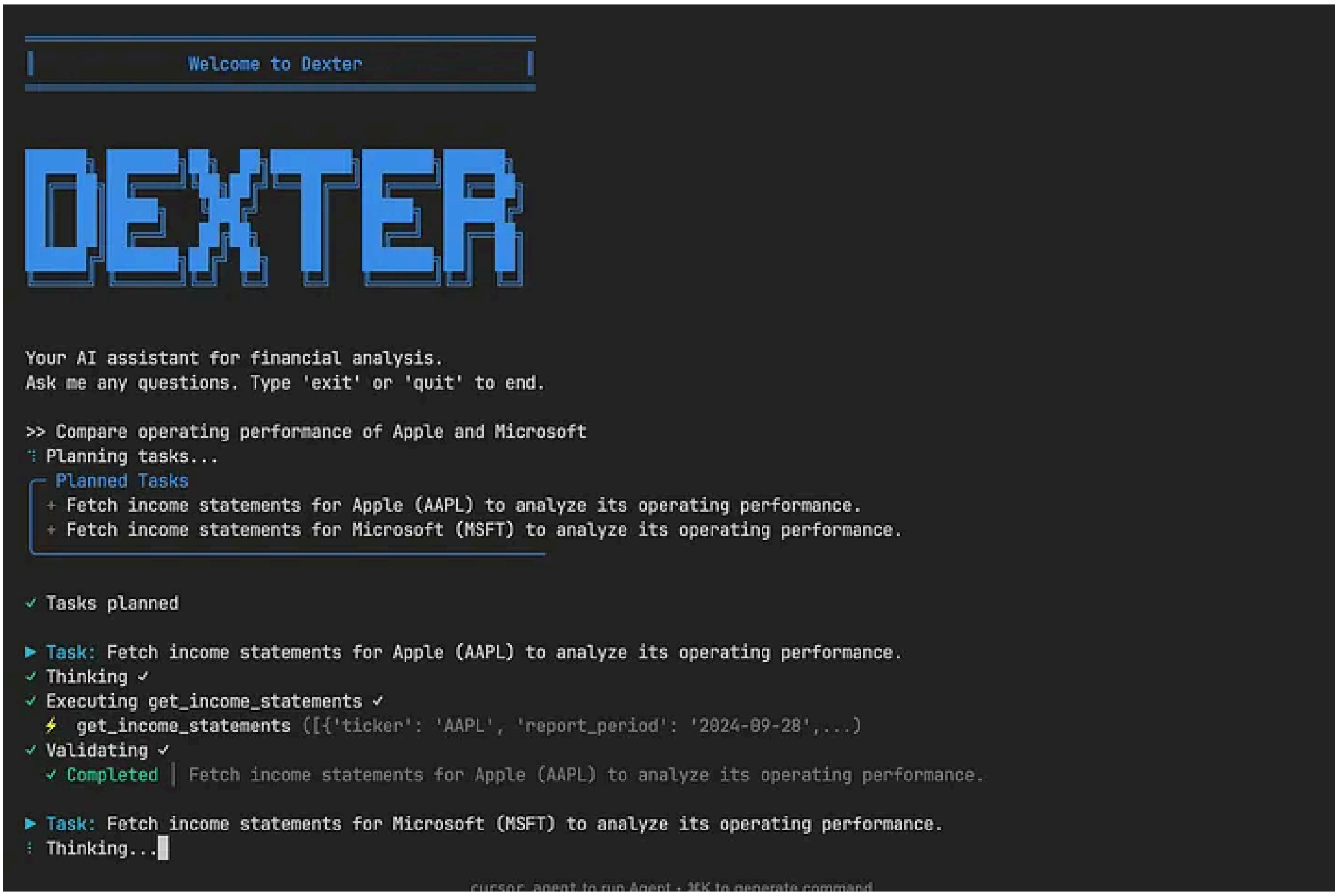
It was 2 AM, and I'd been staring at Apple's quarterly financials for three hours, trying to calculate their revenue growth trajectory across four quarters. My eyes burned from switching between Excel spreadsheets, SEC filings, and financial databases.

There has to be a better way, I thought.

That's when I found Dexter.

Not a person. Not another expensive SaaS tool promising "AI-powered insights" for $299/month. But something different entirely: an autonomous financial research agent built in approximately 200 lines of code that thinks, plans, and validates its own work.

Think Claude Code, but for finance. And it's completely open-source.



```
| Welcome to Dexter |

DEXTER

Your AI assistant for financial analysis.
Ask me any questions. Type 'exit' or 'quit' to end.

>> Compare operating performance of Apple and Microsoft
⁝ Planning tasks...
╭ Planned Tasks
│ + Fetch income statements for Apple (AAPL) to analyze its operating performance.
│ + Fetch income statements for Microsoft (MSFT) to analyze its operating performance.
╰──────────────────────────────────────────

✓ Tasks planned

▶ Task: Fetch income statements for Apple (AAPL) to analyze its operating performance.
✓ Thinking ✓
✓ Executing get_income_statements ✓
  ⚡ get_income_statements ([{'ticker': 'AAPL', 'report_period': '2024-09-28',...)
✓ Validating ✓
  ✓ Completed │ Fetch income statements for Apple (AAPL) to analyze its operating performance.

▶ Task: Fetch income statements for Microsoft (MSFT) to analyze its operating performance.
⁝ Thinking...█
                              cursor agent to run Agent · ⌘K to generate command
```

The Problem Every Investor Faces

Here's the brutal truth about financial research: it's fragmented, time-consuming, and expensive.

Want to compare Microsoft and Google's operating margins? You'll need to:

- Pull income statements from two different sources
- Normalize the data (because companies report differently)
- Build calculations in Excel
- Cross-reference with balance sheets
- Validate your math
- Write up your findings

Repeat this for every stock you research. Every quarter.

Professional analysts have Bloomberg terminals ($24,000/year) and teams of researchers. Individual investors have… Google and hope.

Until now.

Enter Dexter: The AI That Thinks Like an Analyst

Dexter isn't just another financial data API wrapper. It's an autonomous multi-agent system that genuinely reasons through financial problems.

Here's what happened when I asked it a complex question:

"What was Tesla's free cash flow trend over the last 4 quarters, and how does it compare to their capital expenditure growth?"

Most tools would give me raw numbers. Dexter did something remarkable.

First, it broke down my question into discrete research tasks (planning).

Then, it autonomously fetched Tesla's cash flow statements across multiple quarters.

Next, it calculated the trends, performed comparative analysis, and validated its own work.

Finally, it synthesized everything into a comprehensive answer, complete with context about what the numbers actually mean.

Total time: 47 seconds.

The Architecture That Changes Everything

What makes Dexter special isn't just what it does — it's *how* it thinks.

The system orchestrates four specialized AI agents, each with a distinct role :

The Planning Agent acts like your research director. Give it a vague question like "Is Amazon financially healthy?" and it decomposes this into specific, answerable tasks: analyze debt ratios, examine cash flow trends, review revenue growth, assess profitability metrics.

The Action Agent is your tireless analyst. It selects the right tools from its arsenal and executes each research step, interfacing with financial databases that contain 30+ years of data for 16,000+ tickers.

The Validation Agent serves as your fact-checker. After each task completes, it verifies the data is sufficient and accurate before moving forward — no hallucinations, no guesswork.

The Answer Agent becomes your storyteller. It takes all the raw research and weaves it into clear, actionable insights that actually answer your original question.

This isn't prompt engineering magic. It's a genuinely novel approach to autonomous research.

Getting Started: Simpler Than You Think

I'm not a finance expert. I'm an embedded systems engineer who writes about AI tools.

If I can get Dexter running, anyone can.

The Setup (5 Minutes, Seriously)

The beauty of Dexter is its simplicity. You need three things: Python 3.10 or higher, an OpenAI API key, and a Financial Datasets API key.

First, clone the repository to your local machine:

```
git clone https://github.com/virattt/dexter.git
cd dexter
```

Install dependencies using the uv package manager (it handles everything automatically):

```
uv sync
```

Configure your environment by copying the example file and adding your API keys:

```
cp env.example .env
# Edit .env with your OPENAI_API_KEY and FINANCIAL_DATASETS_API_KEY
```

Launch Dexter:

```
uv run dexter-agent
```

That’s it. No Docker containers, no complex configurations, no database setup.

The Financial Datasets Piece

Here’s the only catch: Dexter needs quality financial data to work its magic.

The Financial Datasets API provides access to comprehensive stock market data — income statements, balance sheets, cash flows for 16,000+ tickers spanning three decades. It’s what turns Dexter from a clever chatbot into a genuine research tool.

You’ll need to sign up at financialdatasets.ai and choose a plan (they offer both subscriptions and pay-as-you-go credits). Once you have your API key, add it to your `.env` file and you're ready to research.

Think of it as paying for Bloomberg-quality data at a fraction of the cost.

Real Questions, Real Answers

The moment Dexter clicked for me was when I tested it against my own analysis.

I’d spent an afternoon researching NVIDIA’s revenue growth patterns. When I asked Dexter the same question, it returned an answer that matched my findings — but included context I’d missed about seasonality and segment breakdowns.

Try these queries to see what it can do :

“Compare Microsoft and Google’s operating margins for 2023”

Dexter will pull both companies’ income statements, calculate operating margins, provide the comparison, and add context about which business segments drive the differences.

“What is Amazon’s debt-to-equity ratio based on recent financials?”

It fetches the latest balance sheet, performs the calculation, and explains whether the ratio indicates healthy leverage or potential risk.

“Analyze Tesla’s cash flow trends over the past year”

You'll get quarterly cash flow data, trend analysis, and insights about whether the company is generating or burning cash — critical information for investors.

What Makes This Different

I've tested FinRobot, FinAgent, and other AI financial tools.

Dexter wins on three fronts :

Radical simplicity — The entire codebase is roughly 200 lines. You can actually read and understand how it works. Most frameworks require a PhD to configure.

True autonomy — Dexter doesn't just retrieve data; it plans multi-step research workflows without human intervention. Ask a complex question, and watch it break the problem down, gather evidence, validate findings, and synthesize conclusions.

Transparent reasoning — The multi-agent architecture lets you see exactly how it reached its conclusions. No black box magic, no "trust us" moments.

The Code That Powers It All

For the developers in the room, here's what impressed me most :

The project structure is beautifully organized:

```
dexter/
├── agent.py       # Main orchestration
├── model.py       # LLM interface
├── tools.py       # Financial data tools
├── prompts.py     # Agent system prompts
├── schemas.py     # Data validation
└── cli.py         # Command line interface
```

Each component has a single, clear responsibility. The agent orchestration logic uses LangChain for managing LLM interactions, while Pydantic models ensure data validation and type safety.

You can configure safety limits to prevent runaway execution:

```
agent = Agent(
    max_steps=20,              # Global limit
```

```
        max_steps_per_task=5       # Per-task limit
    )
```

This is production-grade code written for humans to understand.

The Growing Movement

Since its release on October 13, 2025, Dexter has accumulated over 1,300 GitHub stars and 147 forks.

The community is already extending it — adding new data sources, building visualization capabilities, creating sector-specific analysis modules.

One contributor built a portfolio tracking extension. Another added cryptocurrency support. A third created automated report generation.

This is what happens when you build something genuinely useful and make it open-source.

Where This Goes Next

The implications extend beyond individual investors.

Financial analysts could use Dexter as a first-pass research tool, automating the data gathering while they focus on interpretation. Students learning financial analysis get hands-on experience with real data and AI systems. Researchers studying market trends can process datasets at scale.

The codebase is MIT licensed — meaning you can fork it, modify it, build commercial applications on top of it.

Want to add news sentiment analysis? Build it. Need integration with your brokerage API? Add it. Require sector-specific metrics? Extend it.

The Bigger Picture

We're witnessing something significant: the democratization of sophisticated financial analysis.

Tools that cost tens of thousands annually are being replicated — and in some cases, surpassed — by open-source AI agents anyone can run.

Dexter won't replace Bloomberg terminals for institutional investors. But for individual investors, students, analysts at smaller firms, and anyone who needs financial insights without enterprise budgets? This changes the game.

The best part? You can inspect every line of code. You know exactly what it's doing, how it's doing it, and why.

Try It Yourself

The repository is live at [GitHub — virattt/dexter: An autonomous agent for deep financial research]

The setup takes five minutes. The first query you run will feel like magic.

Start with a simple question about a company you're researching. Watch Dexter break it down, gather data, validate its work, and deliver insights.

Then try something more complex. Push the boundaries. See where it excels and where it needs improvement.

And if you build something interesting on top of it? The community would love to see it.

. . .

The future of financial research isn't locked behind expensive paywalls. It's open-source, autonomous, and written in 200 lines of code that anyone can read.

This story is published on Generative AI. Connect with us on LinkedIn and follow Zeniteq to stay in the loop with the latest AI stories.

Subscribe to our newsletter and YouTube channel to stay updated with the latest news and updates on generative AI. Let's shape the future of AI together!

Finance Business Artificial Intelligence Stock Market Software Development

Follow

Published in Generative AI

62K followers · Last published 22 hours ago

Stay updated with the latest news, research, and developments in the world of generative AI. We cover everything from AI model updates, comprehensive tutorials, and real-world applications to the broader impact of AI on society. Work with us: jimclydegm@gmail.com

Following

Written by Adham Khaled

159 followers · 90 following

Embedded Systems Engineer || AI & Tech enthusiast || https://linktr.ee/adhamhidawy

No responses yet

Bgerby

What are your thoughts?

More from Adham Khaled and Generative AI

In AI Mind by Adham Khaled

The Developer’s Co-pilot is Dead: How Factory AI’s Droid Ushers in the Age of Agent-Native…

Forget code completion—Droid autonomously handles entire software development workflows while beating Claude Code and GPT-5 on industry…

✦ Sep 29 👏 66

In Generative AI by Joe Njenga

15 Ways I’m Using Google’s Nano Banana for UI/UX Design (Like a Pro)

I am not a UI/UX designer, but Google Nano Banana is filling the gap, and that doesn't mean I'm becoming one overnight.

✦ Oct 9 512 7

In Generative AI by Dr. GenAI

Hierarchical Reasoning Model (HRM): a tiny brain that embarrasses giant LLMs

HRM is a 27M-parameter model trained in hours that beats giant LLMs on reasoning puzzles, proving architecture can trump scale.

✦ Sep 7 306 6

In Data And Beyond by Adham Khaled

Google’s startup guide to AI agents: ADK, MCP, A2A, and Agentic RAG in practice

✦ Oct 5 👏 54

See all from Adham Khaled

See all from Generative AI

Recommended from Medium

In CodeToDeploy by TechToFit - Master Your Life with Tech

I Tried Google’s New AI Agents. It’s a Gold Rush.

I spend my days deep in the world of AI, but every so often, something drops that makes me stop everything. This is one of those times...

✦ Oct 10 👏 164 💬 3

In The Generator by Thomas Smith

OpenAI Finally Admits the Real Reason it Crippled GPT-5

And what it’s doing to make things right

4d ago 865 27

In AI Software Engineer by Joe Njenga

Why Claude Weekly Limits Are Making Everyone Angry (And $100/Month Plan Will Not Save You)

Yesterday, I finally hit my weekly Claude limit, and I wasn't surprised, since I see dozens of other users online going crazy over these...

4d ago 92 15

In Level Up Coding by Fareed Khan

Building an Agentic Deep-Thinking RAG Pipeline to Solve Complex Queries

Planning, Retrieval, Reflection, Critique, Synthesis and more

✦ 4d ago 👏 801 💬 4

In Dare To Be Better by Max Petrusenko

Claude Skills: The $3 Automation Secret That’s Making Enterprise Teams Look Like Wizards

How a simple folder is replacing $50K consultants and saving companies literal days of work

✦ 6d ago 👏 213 💬 4

Jannis

I Built a Claude Skill in Under 30 Minutes And It Immediately Elevated My Workflow

From zero to skill in 30 minutes: the easiest way to extend Claude with your own workflow.

✦ 5d ago 81

See more recommendations