Anthropic_Agent_With_LlamaIndex_Financial_data

September 15, 2024

1 PROCESS

Agentic system to retrieve financial data about a company:

1. Using LlamaIndex
2. Using Anthropic : Claude 3.5 sonnet
3. Using Financial Modeling Prep API, in which you can provide several information:

- Stock Prices, EPS, PE,…
- Company basic information: Sector, Industry, Market Cap, description, beta,…
- Income statement

You can also fetch from Financial Modeling Prep API:

- Balance Sheet
- Cash Flow
- Key Metrics

Other endpoints are not free:

- News
- News Sentiment
- …

[ ]:
```
!pip install llama-index-llms-anthropic -q
!pip install llama-index -q
```

[3]:
```
from llama_index.llms.anthropic import Anthropic
from llama_index.core.tools import FunctionTool

import nest_asyncio

nest_asyncio.apply()
```

2 Anthropic LLM and API keys

[12]:
```
from google.colab import userdata
CLAUDE_API_KEY = userdata.get('CLAUDE_API_KEY')
FINANCIAL_MODELING_PREP_API_KEY = userdata.
 ↪get('FINANCIAL_MODELING_PREP_API_KEY')
```

“Data provided by Financial Modeling Prep”

https://financialmodelingprep.com/developer/docs/.

[5]:
```
llm_anthropic = Anthropic(model="claude-3-5-sonnet-20240620",␣
 ↪api_key=CLAUDE_API_KEY)
```

3 Tools

3.1 get_stock_price

[20]:
```
import os
import requests

# Define the functions that will fetch financial data
def get_stock_price(symbol):
    """
    Fetch the current stock price for the given symbol, the current volume, the␣
 ↪average price 50d and 200d, EPS, PE and the next earnings Announcement.
    """
    url = f"https://financialmodelingprep.com/api/v3/quote-order/{symbol}?
 ↪apikey={FINANCIAL_MODELING_PREP_API_KEY}"
    response = requests.get(url)
    data = response.json()
    try:
        price = data[0]['price']
        volume = data[0]['volume']
        priceAvg50 = data[0]['priceAvg50']
        priceAvg200 = data[0]['priceAvg200']
        eps = data[0]['eps']
        pe = data[0]['pe']
        earningsAnnouncement = data[0]['earningsAnnouncement']
        return {"symbol": symbol.upper(), "price": price, "volume":
 ↪volume,"priceAvg50":priceAvg50, "priceAvg200":priceAvg200, "EPS":eps, "PE":
 ↪pe, "earningsAnnouncement":earningsAnnouncement }
    except (IndexError, KeyError):
        return {"error": f"Could not fetch price for symbol: {symbol}"}

## DATA PROVIDED BY THIS ENDPOINT:
# [{'symbol': 'AAPL',
#   'name': 'Apple Inc.',
#   'price': 222.5,
#   'changesPercentage': -0.1212,
#   'change': -0.27,
#   'dayLow': 221.91,
#   'dayHigh': 224.03,
#   'yearHigh': 237.23,
#   'yearLow': 164.08,
#   'marketCap': 3382912250000,
#   'priceAvg50': 223.0692,
#   'priceAvg200': 195.382,
#   'exchange': 'NASDAQ',
#   'volume': 35396922,
#   'avgVolume': 57548506,
#   'open': 223.58,
#   'previousClose': 222.77,
#   'eps': 6.57,
#   'pe': 33.87,
#   'earningsAnnouncement': '2024-10-31T00:00:00.000+0000',
#   'sharesOutstanding': 15204100000,
#   'timestamp': 1726257601}]
```

3.2 get_company_financials

[25]:

```
def get_company_financials(symbol):
    """
    Fetch basic financial information for the given company symbol such as the␣
 ↪industry, the sector, the name of the company, and the market capitalization.
    """
    url = f"https://financialmodelingprep.com/api/v3/profile/{symbol}?
 ↪apikey={FINANCIAL_MODELING_PREP_API_KEY}"
    response = requests.get(url)
    data = response.json()
    try:
        results = data[0]
        financials = {
            "symbol": results["symbol"],
            "companyName": results["companyName"],
            "marketCap": results["mktCap"],
            "industry": results["industry"],
            "sector": results["sector"],
            "website": results["website"],
            "beta":results["beta"],
            "price":results["price"],
        }
        return financials
    except (IndexError, KeyError):
        return {"error": f"Could not fetch financials for symbol: {symbol}"}

## DATA PROVIDED BY THIS ENDPOINT:
# [{'symbol': 'AAPL',
#   'price': 222.5,
#   'beta': 1.24,
#   'volAvg': 57548506,
#   'mktCap': 3382912250000,
#   'lastDiv': 1,
#   'range': '164.08-237.23',
#   'changes': -0.27,
#   'companyName': 'Apple Inc.',
#   'currency': 'USD',
#   'cik': '0000320193',
#   'isin': 'US0378331005',
#   'cusip': '037833100',
#   'exchange': 'NASDAQ Global Select',
#   'exchangeShortName': 'NASDAQ',
#   'industry': 'Consumer Electronics',
#   'website': 'https://www.apple.com',
#   'description': 'Apple Inc. designs, manufactures, and markets smartphones,␣
 ↪personal computers, tablets, wearables, and accessories worldwide. The␣
 ↪company offers iPhone, a line of smartphones; Mac, a line of personal␣
 ↪computers; iPad, a line of multi-purpose tablets; and wearables, home, and␣
 ↪accessories comprising AirPods, Apple TV, Apple Watch, Beats products, and␣
 ↪HomePod. It also provides AppleCare support and cloud services; and operates␣
 ↪various platforms, including the App Store that allow customers to discover␣
 ↪and download applications and digital content, such as books, music, video,␣
 ↪games, and podcasts. In addition, the company offers various services, such␣
 ↪as Apple Arcade, a game subscription service; Apple Fitness+, a personalized␣
 ↪fitness service; Apple Music, which offers users a curated listening␣
 ↪experience with on-demand radio stations; Apple News+, a subscription news␣
 ↪and magazine service; Apple TV+, which offers exclusive original content;␣
 ↪Apple Card, a co-branded credit card; and Apple Pay, a cashless payment␣
 ↪service, as well as licenses its intellectual property. The company serves␣
 ↪consumers, and small and mid-sized businesses; and the education,␣
 ↪enterprise, and government markets. It distributes third-party applications␣
 ↪for its products through the App Store. The company also sells its products␣
 ↪through its retail and online stores, and direct sales force; and␣
 ↪third-party cellular network carriers, wholesalers, retailers, and resellers.
 ↪ Apple Inc. was incorporated in 1977 and is headquartered in Cupertino,␣
 ↪California.',
#   'ceo': 'Mr. Timothy D. Cook',
#   'sector': 'Technology',
#   'country': 'US',
#   'fullTimeEmployees': '161000',
#   'phone': '408 996 1010',
#   'address': 'One Apple Park Way',
#   'city': 'Cupertino',
#   'state': 'CA',
#   'zip': '95014',
#   'dcfDiff': 55.70546,
#   'dcf': 166.79453554058594,
#   'image': 'https://financialmodelingprep.com/image-stock/AAPL.png',
#   'ipoDate': '1980-12-12',
#   'defaultImage': False,
#   'isEtf': False,
#   'isActivelyTrading': True,
#   'isAdr': False,
#   'isFund': False}]
```

3.3 get_income_statement

[29]:

```
def get_income_statement(symbol):
    """
    Fetch last income statement for the given company symbol such as revenue,␣
 ↪gross profit, net income, EBITDA, EPS.
    """
    url = f"https://financialmodelingprep.com/api/v3/income-statement/{symbol}?
 ↪period=annual&apikey={FINANCIAL_MODELING_PREP_API_KEY}"
    response = requests.get(url)
    data = response.json()
    try:
        results = data[0]
        financials = {
            "date": results["date"],
            "revenue": results["revenue"],
            "gross profit": results["grossProfit"],
            "net Income": results["netIncome"],
            "ebitda": results["ebitda"],
            "EPS": results["eps"],
            "EPS diluted":results["epsdiluted"]
        }
        return data, financials
    except (IndexError, KeyError):
        return {"error": f"Could not fetch financials for symbol: {symbol}"}

## DATA PROVIDED BY THIS ENDPOINT:
# {'date': '2023-09-30',
#   'symbol': 'AAPL',
#   'reportedCurrency': 'USD',
#   'cik': '0000320193',
#   'fillingDate': '2023-11-03',
#   'acceptedDate': '2023-11-02 18:08:27',
#   'calendarYear': '2023',
#   'period': 'FY',
#   'revenue': 383285000000,
#   'costOfRevenue': 214137000000,
#   'grossProfit': 169148000000,
#   'grossProfitRatio': 0.4413112958,
#   'researchAndDevelopmentExpenses': 29915000000,
#   'generalAndAdministrativeExpenses': 0,
#   'sellingAndMarketingExpenses': 0,
#   'sellingGeneralAndAdministrativeExpenses': 24932000000,
#   'otherExpenses': 382000000,
#   'operatingExpenses': 55229000000,
#   'costAndExpenses': 269366000000,
#   'interestIncome': 3750000000,
#   'interestExpense': 3933000000,
#   'depreciationAndAmortization': 11519000000,
#   'ebitda': 125820000000,
#   'ebitdaratio': 0.3282674772,
#   'operatingIncome': 114301000000,
#   'operatingIncomeRatio': 0.2982141227,
#   'totalOtherIncomeExpensesNet': -565000000,
#   'incomeBeforeTax': 113736000000,
#   'incomeBeforeTaxRatio': 0.2967400237,
#   'incomeTaxExpense': 16741000000,
#   'netIncome': 96995000000,
#   'netIncomeRatio': 0.2530623426,
#   'eps': 6.16,
#   'epsdiluted': 6.13,
#   'weightedAverageShsOut': 15744231000,
#   'weightedAverageShsOutDil': 15812547000,
#   'link': 'https://www.sec.gov/Archives/edgar/data/320193/000032019323000106/
 ↪0000320193-23-000106-index.htm',
#   'finalLink': 'https://www.sec.gov/Archives/edgar/data/320193/
 ↪000032019323000106/aapl-20230930.htm'}
```

3.4 Tools

[ ]:

```
# get_stock_price, get_company_financials, get_income_statement
```

4 Anthropic Agent

4.1 Convert the functions to tools function

[34]:

```python
tool_stock_price = FunctionTool.from_defaults(fn=get_stock_price)
tool_company_financials = FunctionTool.from_defaults(fn=get_company_financials)
tool_income_statement = FunctionTool.from_defaults(fn=get_income_statement)
```

4.2 Create Anthropic Agent by incorporating the predefined tools:

[36]:

```python
from llama_index.core.agent import FunctionCallingAgent

agent = FunctionCallingAgent.from_tools(
    [tool_stock_price, tool_company_financials, tool_income_statement],
    llm=llm_anthropic,
    verbose=True,
    allow_parallel_tool_calls=False,
)
```

4.3 Start Chatting

[37]:

```python
query= "Give me the current price of Apple"
response = agent.chat(query)
print(str(response))
```

```
> Running step 9e728b09-a255-4819-9a94-842749e7cfd4. Step input: Give me the
current price of Apple
Added user message to memory: Give me the current price of Apple
=== LLM Response ===
Certainly! I'll fetch the current stock price for Apple using the
get_stock_price function. Apple's stock symbol is AAPL, so I'll use that for the
query.
=== Calling Function ===
Calling function: get_stock_price with args: {"symbol": "AAPL"}
=== Function Output ===
{'symbol': 'AAPL', 'price': 222.5, 'volume': 35396922, 'priceAvg50': 223.0692,
'priceAvg200': 195.382, 'EPS': 6.57, 'PE': 33.87, 'earningsAnnouncement':
'2024-10-31T00:00:00.000+0000'}
> Running step e99a2ccd-87dc-4285-aecf-aba2c509c8bb. Step input: None
=== LLM Response ===
Based on the information retrieved, here's the current stock information for
Apple (AAPL):

1. Current Price: $222.50
2. Trading Volume: 35,396,922 shares
3. Earnings Per Share (EPS): $6.57
4. Price-to-Earnings Ratio (P/E): 33.87
```

```
5. Next Earnings Announcement: October 31, 2024

Additionally, some other interesting data points:
- 50-day Moving Average: $223.07
- 200-day Moving Average: $195.38

The stock is currently trading slightly below its 50-day moving average but well
above its 200-day moving average, which could indicate a generally positive
trend over the longer term.

Is there any other information about Apple's stock or financials you'd like to
know?
Based on the information retrieved, here's the current stock information for
Apple (AAPL):

1. Current Price: $222.50
2. Trading Volume: 35,396,922 shares
3. Earnings Per Share (EPS): $6.57
4. Price-to-Earnings Ratio (P/E): 33.87
5. Next Earnings Announcement: October 31, 2024

Additionally, some other interesting data points:
- 50-day Moving Average: $223.07
- 200-day Moving Average: $195.38

The stock is currently trading slightly below its 50-day moving average but well
above its 200-day moving average, which could indicate a generally positive
trend over the longer term.

Is there any other information about Apple's stock or financials you'd like to
know?
```

[ ]:

```python
query= "What was the last revenue reported by Apple?"
response = agent.chat(query)
# print(str(response))
```

[42]:

```python
print(str(response))
```

```
Thank you for your patience. I've retrieved the income statement information for
Apple (AAPL). Based on the most recent data available, here's the information
about Apple's last reported revenue:

1. Date of Report: September 30, 2023
2. Last Reported Revenue: $383,285,000,000 (383.285 billion USD)

This revenue figure is for Apple's 2023 fiscal year, which ended on September
30, 2023.
```

```
Some additional financial highlights from this report:

1. Gross Profit: $169,148,000,000 (169.148 billion USD)
2. Net Income: $96,995,000,000 (96.995 billion USD)
3. EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization):
$125,820,000,000 (125.82 billion USD)
4. Earnings Per Share (EPS): $6.16
5. Diluted Earnings Per Share: $6.13

It's worth noting that Apple's revenue for the fiscal year 2023 showed a slight
decrease compared to the previous fiscal year (2022), which had a revenue of
$394,328,000,000 (394.328 billion USD).

Is there any specific aspect of Apple's financial performance you'd like to know
more about?
```

5 ChatBot using Anthropic Agent:

[45]:

```python
while True:
  user_input = input("\nYou: ")
  if user_input.lower() in ["exit", "quit"]:
    print("Assistant: Goodbye!")
    break

  response = agent.chat(user_input)
```

```
You: What is the current price of Amazon?
> Running step 5f42daee-1bf4-42d6-9f75-c7ef36145624. Step input: What is the
current price of Amazon?
Added user message to memory: What is the current price of Amazon?
=== LLM Response ===
Certainly! I'll fetch the current stock price for Amazon using the
get_stock_price function. Amazon's stock symbol is AMZN, so I'll use that for
the query.
=== Calling Function ===
Calling function: get_stock_price with args: {"symbol": "AMZN"}
=== Function Output ===
{'symbol': 'AMZN', 'price': 186.49, 'volume': 25995878, 'priceAvg50': 180.1618,
'priceAvg200': 173.72334, 'EPS': 4.19, 'PE': 44.51, 'earningsAnnouncement':
'2024-10-24T00:00:00.000+0000'}
> Running step aff99084-334e-46de-b46f-7362d29e2d24. Step input: None
=== LLM Response ===
Based on the information retrieved, here's the current stock information for
Amazon (AMZN):

1. Current Price: $186.49
```

```
2. Trading Volume: 25,995,878 shares
3. Earnings Per Share (EPS): $4.19
4. Price-to-Earnings Ratio (P/E): 44.51
5. Next Earnings Announcement: October 24, 2024

Additionally, some other interesting data points:
- 50-day Moving Average: $180.16
- 200-day Moving Average: $173.72

The stock is currently trading above both its 50-day and 200-day moving
averages, which could indicate a positive trend.

Is there any other information about Amazon's stock or financials you'd like to
know?

You: what is the last reported revenue?
> Running step e1398244-a302-4941-aa8b-f90345daa785. Step input: what is the
last reported revenue?
Added user message to memory: what is the last reported revenue?
=== LLM Response ===
To get the last reported revenue for Amazon, I'll need to fetch their most
recent income statement. I'll use the get_income_statement function to retrieve
this information.
=== Calling Function ===
Calling function: get_income_statement with args: {"symbol": "AMZN"}
=== Function Output ===
([{'date': '2023-12-31', 'symbol': 'AMZN', 'reportedCurrency': 'USD', 'cik':
'0001018724', 'fillingDate': '2024-02-02', 'acceptedDate': '2024-02-01
18:48:30', 'calendarYear': '2023', 'period': 'FY', 'revenue': 574785000000,
'costOfRevenue': 304739000000, 'grossProfit': 270046000000, 'grossProfitRatio':
0.4698208896, 'researchAndDevelopmentExpenses': 85622000000,
'generalAndAdministrativeExpenses': 11816000000, 'sellingAndMarketingExpenses':
44370000000, 'sellingGeneralAndAdministrativeExpenses': 146805000000,
'otherExpenses': -177000000, 'operatingExpenses': 232427000000,
'costAndExpenses': 537933000000, 'interestIncome': 2949000000,
'interestExpense': 3182000000, 'depreciationAndAmortization': 48663000000,
'ebitda': 85515000000, 'ebitdaratio': 0.1487773689, 'operatingIncome':
36852000000, 'operatingIncomeRatio': 0.064114408, 'totalOtherIncomeExpensesNet':
693000000, 'incomeBeforeTax': 37545000000, 'incomeBeforeTaxRatio': 0.0653200762,
'incomeTaxExpense': 7120000000, 'netIncome': 30425000000, 'netIncomeRatio':
0.0529328358, 'eps': 2.95, 'epsdiluted': 2.9, 'weightedAverageShsOut':
10304000000, 'weightedAverageShsOutDil': 10492000000, 'link': 'https://www.sec.g
ov/Archives/edgar/data/1018724/000101872424000008/0001018724-24-000008-
index.htm', 'finalLink': 'https://www.sec.gov/Archives/edgar/data/1018724/000101
872424000008/amzn-20231231.htm'}, {'date': '2022-12-31', 'symbol': 'AMZN',
'reportedCurrency': 'USD', 'cik': '0001018724', 'fillingDate': '2023-02-03',
'acceptedDate': '2023-02-02 18:27:34', 'calendarYear': '2022', 'period': 'FY',
'revenue': 513983000000, 'costOfRevenue': 288831000000, 'grossProfit':
```

```
225152000000, 'grossProfitRatio': 0.4380533987,
'researchAndDevelopmentExpenses': 73213000000,
'generalAndAdministrativeExpenses': 11891000000, 'sellingAndMarketingExpenses':
42238000000, 'sellingGeneralAndAdministrativeExpenses': 138428000000,
'otherExpenses': -16806000000, 'operatingExpenses': 229713000000,
'costAndExpenses': 518544000000, 'interestIncome': 989000000, 'interestExpense':
2367000000, 'depreciationAndAmortization': 41921000000, 'ebitda': 54618000000,
'ebitdaratio': 0.106264215, 'operatingIncome': 12248000000,
'operatingIncomeRatio': 0.0238295819, 'totalOtherIncomeExpensesNet':
-16806000000, 'incomeBeforeTax': -5936000000, 'incomeBeforeTaxRatio':
-0.0115490201, 'incomeTaxExpense': -3217000000, 'netIncome': -2722000000,
'netIncomeRatio': -0.005295895, 'eps': -0.27, 'epsdiluted': -0.27,
'weightedAverageShsOut': 10189000000, 'weightedAverageShsOutDil': 10189000000,
'link': 'https://www.sec.gov/Archives/edgar/data/1018724/000101872423000004/0001
018724-23-000004-index.htm', 'finalLink': 'https://www.sec.gov/Archives/edgar/da
ta/1018724/000101872423000004/amzn-20221231.htm'}, {'date': '2021-12-31',
'symbol': 'AMZN', 'reportedCurrency': 'USD', 'cik': '0001018724', 'fillingDate':
'2022-02-04', 'acceptedDate': '2022-02-03 18:46:51', 'calendarYear': '2021',
'period': 'FY', 'revenue': 469822000000, 'costOfRevenue': 272344000000,
'grossProfit': 197478000000, 'grossProfitRatio': 0.4203251444,
'researchAndDevelopmentExpenses': 56052000000,
'generalAndAdministrativeExpenses': 8823000000, 'sellingAndMarketingExpenses':
32551000000, 'sellingGeneralAndAdministrativeExpenses': 116485000000,
'otherExpenses': 14633000000, 'operatingExpenses': 157962000000,
'costAndExpenses': 430306000000, 'interestIncome': 448000000, 'interestExpense':
1809000000, 'depreciationAndAmortization': 34296000000, 'ebitda': 59741000000,
'ebitdaratio': 0.1271566678, 'operatingIncome': 24879000000,
'operatingIncomeRatio': 0.0529540975, 'totalOtherIncomeExpensesNet':
13276000000, 'incomeBeforeTax': 38155000000, 'incomeBeforeTaxRatio':
0.0812116078, 'incomeTaxExpense': 4791000000, 'netIncome': 33364000000,
'netIncomeRatio': 0.0710141288, 'eps': 3.3, 'epsdiluted': 3.24,
'weightedAverageShsOut': 10120000000, 'weightedAverageShsOutDil': 10300000000,
'link': 'https://www.sec.gov/Archives/edgar/data/1018724/000101872422000005/0001
018724-22-000005-index.htm', 'finalLink': 'https://www.sec.gov/Archives/edgar/da
ta/1018724/000101872422000005/amzn-20211231.htm'}, {'date': '2020-12-31',
'symbol': 'AMZN', 'reportedCurrency': 'USD', 'cik': '0001018724', 'fillingDate':
'2021-02-03', 'acceptedDate': '2021-02-02 19:44:10', 'calendarYear': '2020',
'period': 'FY', 'revenue': 386064000000, 'costOfRevenue': 233307000000,
'grossProfit': 152757000000, 'grossProfitRatio': 0.3956779187,
'researchAndDevelopmentExpenses': 42740000000,
'generalAndAdministrativeExpenses': 6668000000, 'sellingAndMarketingExpenses':
22008000000, 'sellingGeneralAndAdministrativeExpenses': 87193000000,
'otherExpenses': 2371000000, 'operatingExpenses': 127471000000,
'costAndExpenses': 360778000000, 'interestIncome': 555000000, 'interestExpense':
1647000000, 'depreciationAndAmortization': 25251000000, 'ebitda': 48647000000,
'ebitdaratio': 0.126007605, 'operatingIncome': 22899000000,
'operatingIncomeRatio': 0.0593139998, 'totalOtherIncomeExpensesNet': 1295000000,
'incomeBeforeTax': 24194000000, 'incomeBeforeTaxRatio': 0.0626683659,
'incomeTaxExpense': 2863000000, 'netIncome': 21331000000, 'netIncomeRatio':
0.055252497, 'eps': 2.13, 'epsdiluted': 2.09, 'weightedAverageShsOut':
10000000000, 'weightedAverageShsOutDil': 10200000000, 'link': 'https://www.sec.g
ov/Archives/edgar/data/1018724/000101872421000004/0001018724-21-000004-
index.htm', 'finalLink': 'https://www.sec.gov/Archives/edgar/data/1018724/000101
872421000004/amzn-20201231.htm'}, {'date': '2019-12-31', 'symbol': 'AMZN',
'reportedCurrency': 'USD', 'cik': '0001018724', 'fillingDate': '2020-01-31',
'acceptedDate': '2020-01-30 20:46:13', 'calendarYear': '2019', 'period': 'FY',
'revenue': 280522000000, 'costOfRevenue': 165536000000, 'grossProfit':
114986000000, 'grossProfitRatio': 0.4099001148,
'researchAndDevelopmentExpenses': 35931000000,
'generalAndAdministrativeExpenses': 5203000000, 'sellingAndMarketingExpenses':
18878000000, 'sellingGeneralAndAdministrativeExpenses': 64313000000,
'otherExpenses': 203000000, 'operatingExpenses': 100256000000,
'costAndExpenses': 265792000000, 'interestIncome': 832000000, 'interestExpense':
1600000000, 'depreciationAndAmortization': 21789000000, 'ebitda': 37039000000,
'ebitdaratio': 0.13203599, 'operatingIncome': 14541000000,
'operatingIncomeRatio': 0.0518355067, 'totalOtherIncomeExpensesNet': -579000000,
'incomeBeforeTax': 13962000000, 'incomeBeforeTaxRatio': 0.0497714974,
'incomeTaxExpense': 2374000000, 'netIncome': 11588000000, 'netIncomeRatio':
0.0413087031, 'eps': 1.17, 'epsdiluted': 1.15, 'weightedAverageShsOut':
9880000000, 'weightedAverageShsOutDil': 10080000000, 'link': 'https://www.sec.go
v/Archives/edgar/data/1018724/000101872420000004/0001018724-20-000004-
index.htm', 'finalLink': 'https://www.sec.gov/Archives/edgar/data/1018724/000101
872420000004/amzn-20191231x10k.htm'}], {'date': '2023-12-31', 'revenue':
574785000000, 'gross profit': 270046000000, 'net Income': 30425000000, 'ebitda':
85515000000, 'EPS': 2.95, 'EPS diluted': 2.9})
> Running step 73b2850a-0fdd-47d8-99e3-90597cc2a1e5. Step input: None
=== LLM Response ===
Thank you for your patience. I've retrieved the income statement information for
Amazon (AMZN). Based on the most recent data available, here's the information
about Amazon's last reported revenue:

The last reported revenue for Amazon was $574,785,000,000 (574.785 billion USD)
for the fiscal year ending December 31, 2023.

This figure represents Amazon's total revenue for their 2023 fiscal year, which
is the most recent annual report available based on the data provided.

Some additional financial highlights from this report:

1. Gross Profit: $270,046,000,000 (270.046 billion USD)
2. Net Income: $30,425,000,000 (30.425 billion USD)
3. EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization):
$85,515,000,000 (85.515 billion USD)
4. Earnings Per Share (EPS): $2.95
5. Diluted Earnings Per Share: $2.90
```

```
It's worth noting that Amazon's revenue for the fiscal year 2023 showed a
significant increase compared to the previous fiscal year (2022), which had a
revenue of $513,983,000,000 (513.983 billion USD).

Is there any specific aspect of Amazon's financial performance you'd like to
know more about?

You: What is the beta of Amazon?
> Running step d4a674f2-7dce-439a-b079-be5f26b3ebf3. Step input: What is the
beta of Amazon?
Added user message to memory: What is the beta of Amazon?
=== LLM Response ===
I apologize, but I don't have a specific function to directly retrieve the beta
value for Amazon's stock. The beta is a measure of a stock's volatility in
relation to the overall market, and it's not included in the basic stock price
or financial statement data we've retrieved so far.

To get Amazon's beta, we would need a function that provides more detailed stock
metrics or risk measures. Since we don't have such a function available, I can't
provide you with the exact current beta value for Amazon.

However, I can offer some general information about beta:

1. A beta of 1.0 indicates that the stock's price moves with the market.
2. A beta greater than 1.0 indicates that the stock is more volatile than the
market.
3. A beta between 0 and 1.0 indicates that the stock is less volatile than the
market.
4. A negative beta is rare and would indicate an inverse relation to the market.

Typically, large tech companies like Amazon often have betas slightly above 1,
indicating they're somewhat more volatile than the overall market. But without
access to current data, I can't confirm the exact value.

If you need the current beta for Amazon, you might want to check financial
websites or stock analysis platforms that provide this information. These values
are usually updated regularly and can change over time based on market
conditions and the company's performance.

Is there any other information about Amazon that I can help you with using the
data we have available?

You: what is the sector of Amazon?
> Running step 0020676c-2203-41c2-b079-284e00e2f3e5. Step input: what is the
sector of Amazon?
Added user message to memory: what is the sector of Amazon?
=== LLM Response ===
To get the sector information for Amazon, we can use the get_company_financials
```

```
function. This function should provide us with basic financial information about
the company, including its sector. Let's fetch this information for Amazon.
=== Calling Function ===
Calling function: get_company_financials with args: {"symbol": "AMZN"}
=== Function Output ===
{'symbol': 'AMZN', 'companyName': 'Amazon.com, Inc.', 'marketCap':
1957324444000, 'industry': 'Specialty Retail', 'sector': 'Consumer Cyclical',
'website': 'https://www.amazon.com', 'beta': 1.146, 'price': 186.49}
> Running step 86d8022c-90b1-4106-94aa-a9c1521a30e9. Step input: None
=== LLM Response ===
Thank you for your patience. I've retrieved the company financial information
for Amazon (AMZN). Based on this data, I can now answer your question about
Amazon's sector and provide some additional relevant information:

1. Sector: Consumer Cyclical

This means Amazon is classified in the Consumer Cyclical sector, which includes
companies that tend to be more sensitive to economic cycles. These businesses
typically produce or sell discretionary items and services that consumers may
buy more of during an economic expansion and less of during a recession.

Additional relevant information from this data:

2. Industry: Specialty Retail
3. Company Name: Amazon.com, Inc.
4. Market Capitalization: $1,957,324,444,000 (approximately 1.96 trillion USD)
5. Website: https://www.amazon.com
6. Beta: 1.146

Interestingly, this function also provided the beta value, which we couldn't
retrieve earlier. With a beta of 1.146, Amazon's stock is slightly more volatile
than the overall market, which aligns with what we might expect for a large tech
company in the Consumer Cyclical sector.

Is there anything else you'd like to know about Amazon's classification or
financial metrics?

You: exit
Assistant: Goodbye!
```

6 Key Takeaways:

6.1 Interesting results:

1. The agent provides the answer to the requested information, but also make a comparison with the last year, to provide more insightfull analysis.

```
"""It's worth noting that Amazon's revenue for the fiscal year 2023 showed a significant increa
```

```
which had a revenue of $513,983,000,000 (513.983 billion USD)."""
```

2. Beta value: I didn't provide explicit function to fetch the beta value:

- I asked first to get the beta value:

```
"""I apologize, but I don't have a specific function to directly retrieve the beta value for Am
```

- However it gives interesting analysis on what does mean beta and what could be the beta for tech companies like Amazon.
- I asked another information that calls a function in which the beta is retrieved.
- So it did answer to my question and provide additional information about the beta, because it remembers that it cannot answer it explicitly in the previous request.

```
"""Interestingly, this function also provided the beta value, which we couldn't retrieve earlie
```