

Develop a Reliable Backend with Node and Express



Phase-End Project





Stock API Access Using Express JS

Objective

You are asked to develop a comprehensive and dynamic online stock trading platform that effectively serves individual investors and financial professionals.

This platform will integrate real-time market data, advanced analytics, and personalized portfolio management tools to meet the evolving demands of the modern financial landscape.



Problem Statement and Motivation

Real-time scenario:

In the evolving landscape of financial markets, individual investors and financial advisors face challenges in accessing real-time stock data and analytics. The Stock API Access Using Express JS project aims to bridge this gap by creating a robust online platform for real-time stock market data analysis.

As a backend developer, your primary responsibility is to develop a reliable and efficient system that integrates with stock market APIs. This includes building a backend architecture capable of handling large volumes of data, ensuring secure and seamless data flow, and implementing functionalities for stock data retrieval and analysis. The goal is to provide a comprehensive tool that empowers users with timely market insights, aiding in informed investment decisions.



Industry Relevance



The following tools are widely used in the industry for backend development:

1. **Node.js:** It is crucial for building scalable network applications, particularly in real-time data-intensive environments like stock market analytics.
2. **Express.js:** It is a web application framework for Node.js. It simplifies the development of web APIs, providing the backbone for handling requests and routing.
3. **MongoDB:** This NoSQL database is used for storing large volumes of data efficiently, vital for handling complex financial datasets in the stock market domain.

Tasks

The following tasks outline the backend development process of a Stock API :

1. Initialize a Git repository to track and manage the backend development process
2. Establish the server-side logic and handle API requests and responses using Node.js
3. Implement Express.js for routing and middleware functions, streamlining the interaction with stock market APIs
4. Integrate MongoDB for data storage and retrieval to handle large sets of stock market data
5. Make HTTP requests to external stock market APIs, ensuring seamless data fetching using Axios



Project References

- **Task 1:** Course 1 Lesson 2
- **Task 2:** Course 4 Lesson 5
- **Task 3:** Course 4 Lesson 3
- **Task 4:** Course 4 Lesson 8
- **Task 5:** Course 2 Lesson 7

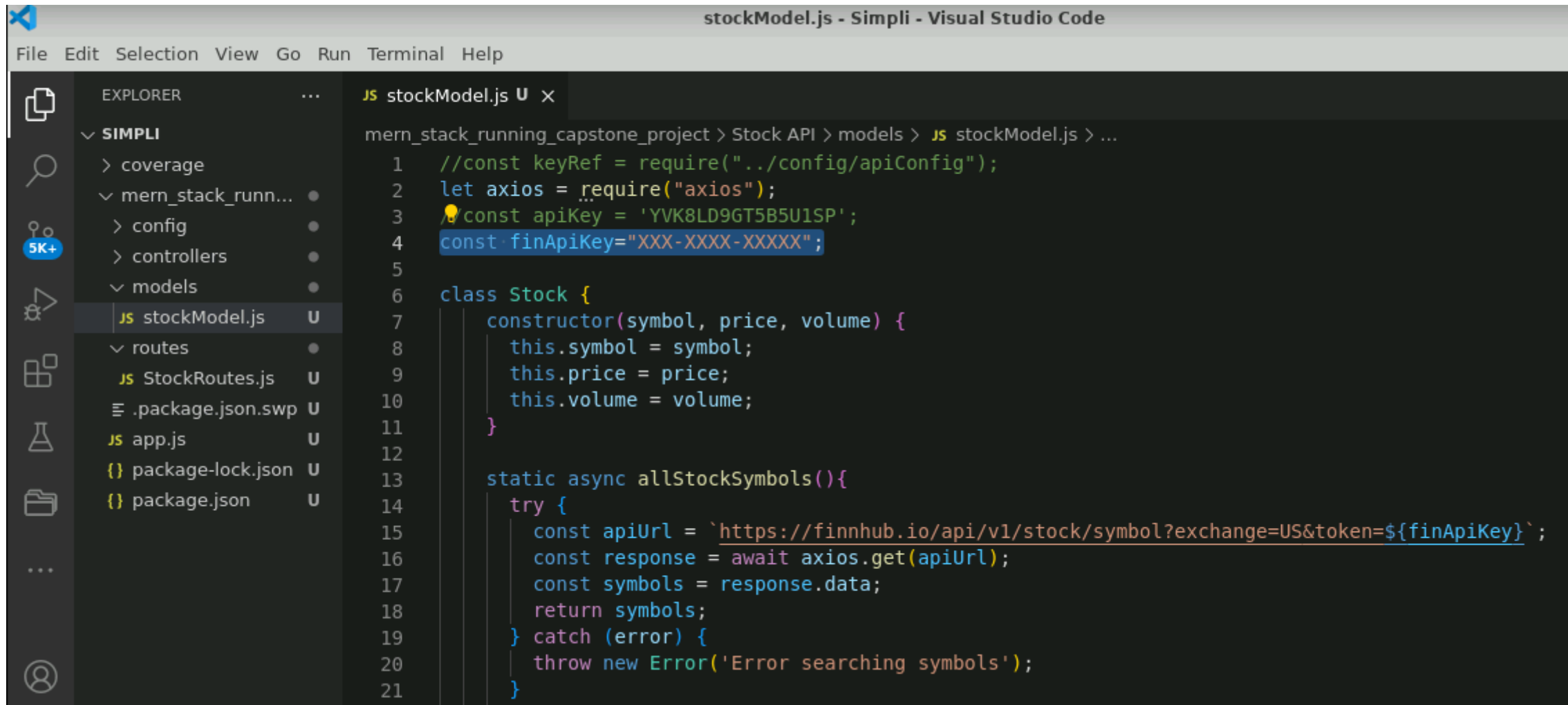


Sample Output

```
1: labuser@ubuntu2204: ~/Simpli ▼  
labuser@ubuntu2204:~/Simpli$ git clone https://github.com/Kaleakash/mern_stack_running_capstone_project.git  
Cloning into 'mern_stack_running_capstone_project'...  
remote: Enumerating objects: 5575, done.  
remote: Total 5575 (delta 0), reused 0 (delta 0), pack-reused 5575  
Receiving objects: 100% (5575/5575), 126.11 MiB | 15.50 MiB/s, done.  
Resolving deltas: 100% (1417/1417), done.  
labuser@ubuntu2204:~/Simpli$
```

```
1: labuser@ubuntu2204: ~/Desktop/Stock API ▼  
labuser@ubuntu2204:~/Desktop/Stock API$ npm install  
  
added 69 packages, and audited 70 packages in 1s  
  
12 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
labuser@ubuntu2204:~/Desktop/Stock API$
```

Sample Output

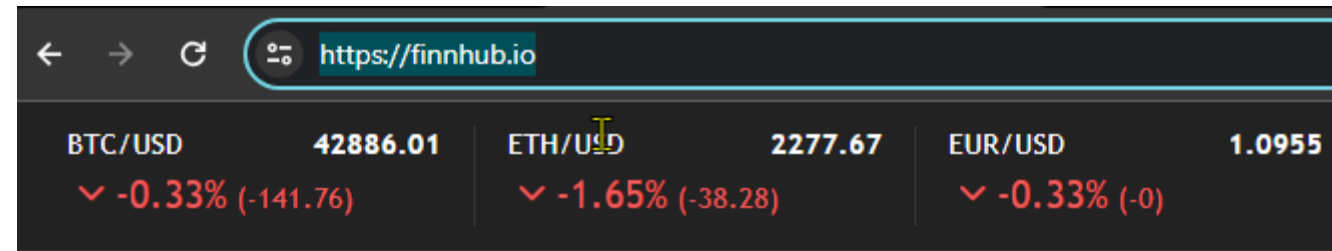


```
stockModel.js - Simpli - Visual Studio Code
File Edit Selection View Go Run Terminal Help

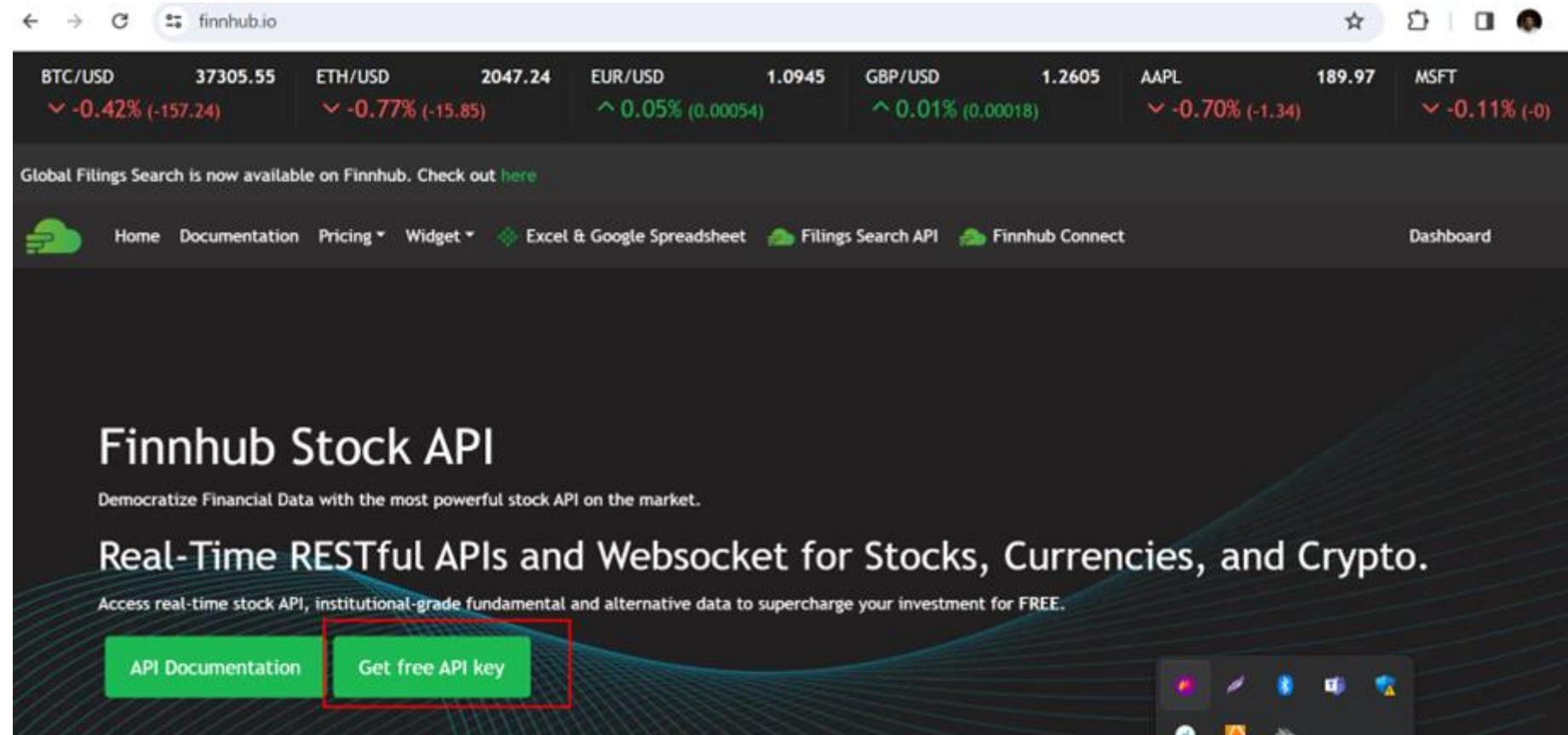
EXPLORER
SIMPLI
  coverage
  mern_stack_runn...
  config
  controllers
  models
    JS stockModel.js
  routes
    JS StockRoutes.js
  .package.json.swp
  app.js
  package-lock.json
  package.json

JS stockModel.js
1 //const keyRef = require("../config/apiConfig");
2 let axios = require("axios");
3 const apiKey = 'YVK8LD9GT5B5U1SP';
4 const finApiKey="XXX-XXXX-XXXXX";
5
6 class Stock {
7   constructor(symbol, price, volume) {
8     this.symbol = symbol;
9     this.price = price;
10    this.volume = volume;
11  }
12
13  static async allStockSymbols(){
14    try {
15      const apiUrl = `https://finnhub.io/api/v1/stock/symbol?exchange=US&token=${finApiKey}`;
16      const response = await axios.get(apiUrl);
17      const symbols = response.data;
18      return symbols;
19    } catch (error) {
20      throw new Error('Error searching symbols');
21    }
22  }
```

Sample Output



BTC/USD	42886.01	ETH/USD	2277.67	EUR/USD	1.0955
▼ -0.33% (-141.76)		▼ -1.65% (-38.28)		▼ -0.33% (-0)	



Global Filings Search is now available on Finnhub. Check out [here](#)

Home Documentation Pricing Widget Excel & Google Spreadsheet Filings Search API Finnhub Connect Dashboard

Finnhub Stock API

Democratize Financial Data with the most powerful stock API on the market.

Real-Time RESTful APIs and Websocket for Stocks, Currencies, and Crypto.

Access real-time stock API, institutional-grade fundamental and alternative data to supercharge your investment for FREE.

[API Documentation](#) [Get free API key](#)

Sample Output

```
clu1sp1r01qlu9ol3kdgclu1sp1r01qlu9ol3ke0
```



```
JS stockModel.js ×
models > JS stockModel.js > finApiKey
1 //const keyRef = require("../config/apiConfig");
2 let axios = require("axios");
3 //const apiKey = 'YVK8LD9GT5B5U1SP';
4 const finApiKey="clu1sp1r01qlu9ol3kdgclu1sp1r01qlu9ol3ke0";
5
6 class Stock {
```

```
labuser@ubuntu2204:~/Desktop/Stock API$ node app.js
Server is running on port 3000
```

localhost:3000/api/stock/all × +

localhost:3000/api/stock/all

```
[{"currency":"USD","description":"EVONIK INDUSTRIES-UNSP
ADR","displaySymbol":"EVKIY","figi":"BBG0084GVF31","isin":null,"mic":"00TC","shareClassFIGI":"BBG0084GVFV0","symbol":"EVKIY","symbol2":"","type":"ADR"},
{"currency":"USD","description":"PAPAYA GROWTH OPPORTUNI-
CW23","displaySymbol":"PPYAW","figi":"BBG015RGK795","isin":null,"mic":"XNAS","shareClassFIGI":"","symbol":"PPYAW","symbol2":"","type":"Equity WRT"},
{"currency":"USD","description":"NEW WORLD GOLD
CORP","displaySymbol":"NWGC","figi":"BBG000D2M3X3","isin":null,"mic":"00TC","shareClassFIGI":"BBG001SHBKS2","symbol":"NWGC","symbol2":"","type":"Common Stock"},
{"currency":"USD","description":"NOWAUTO
INC","displaySymbol":"NWAU","figi":"BBG000Q1J686","isin":null,"mic":"00TC","shareClassFIGI":"BBG001SM28D4","symbol":"NWAU","symbol2":"","type":"Common Stock"},
{"currency":"USD","description":"ALPHA PARTNERS TECHNOLOGY
ME","displaySymbol":"APTMU","figi":"BBG00Z9FP0Q3","isin":null,"mic":"XNAS","shareClassFIGI":"BBG00Z9FP0T0","symbol":"APTMU","symbol2":"","type":"Unit"},
{"currency":"USD","description":"GLOBAL X MSCI CHINA
FINANCIA","displaySymbol":"CHIX","figi":"BBG000Q0K2X7","isin":null,"mic":"ARCX","shareClassFIGI":"BBG001T670Z6","symbol":"CHIX","symbol2":"","type":"ETP"},
{"currency":"USD","description":"BROOKSTONE ULTRASHRT BND
ETF","displaySymbol":"BAMU","figi":"BBG01JG4D4D4","isin":null,"mic":"BATS","shareClassFIGI":"BBG01JG4D578","symbol":"BAMU","symbol2":"","type":"ETP"},
{"currency":"USD","description":"PACER INDUSTRIAL REAL
ESTATE","displaySymbol":"INDS","figi":"BBG00KX8XVM2","isin":null,"mic":"ARCX","shareClassFIGI":"BBG00KX8XWB2","symbol":"INDS","symbol2":"","type":"ETP"},
{"currency":"USD","description":"WQN
INC","displaySymbol":"WQNI","figi":"BBG000BZ8FY0","isin":null,"mic":"00TC","shareClassFIGI":"BBG001SD0DX3","symbol":"WQNI","symbol2":"","type":"Common Stock"},
{"currency":"USD","description":"SPDR S&P KENSHO NEW
ECONOMIE","displaySymbol":"KOMP","figi":"BBG00M9BL6G3","isin":null,"mic":"ARCX","shareClassFIGI":"BBG00M9BL771","symbol":"KOMP","symbol2":"","type":"ETP"},
{"currency":"USD","description":"LION CORP-UNSPONSORED
ADR","displaySymbol":"LIOPY","figi":"BBG001B282Q9","isin":null,"mic":"00TC","shareClassFIGI":"BBG001THLV33","symbol":"LIOPY","symbol2":"","type":"ADR"},
{"currency":"USD","description":"SOFI WEEKLY DIVIDEND
ETF","displaySymbol":"WKLY","figi":"BBG010Z8G059","isin":null,"mic":"ARCX","shareClassFIGI":"BBG010Z8G102","symbol":"WKLY","symbol2":"","type":"ETP"},
```



Thank You