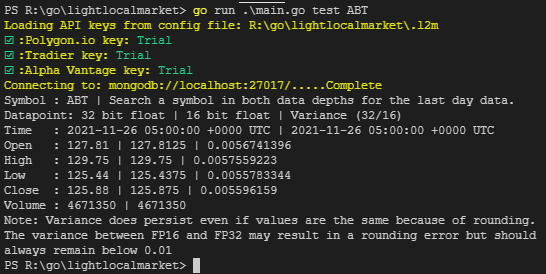
**lightlocalmarket User Guide**

**Overview:**

This user guide explains the procedures for setup and operation of the “lightlocalmarket” integration. Lightlocalmarket automates the aggregation of open-source stock market data into a historical database of daily stock price data for analysis.

This command-line interface program downloads from two public APIs; Alpha Vantage and Tradier. Once retrieved, this data is deposited into a MongoDB with a focus on eliminating most redundant data. This priority is exempted with a complete duplication at lower resolution. Each of the S&P 500 index components generates a database with two collections of daily data.

Due to other significant savings, having one collection in 16-bit floating-point and one 32-bit allows for later optimization computationally without sacrificing significant space requirements. The download should complete in about 90 minutes, but that time-frame is largely due to pace restrictions on API requests and not data size. These data types can be compared using the ‘test [Symbol]` function.

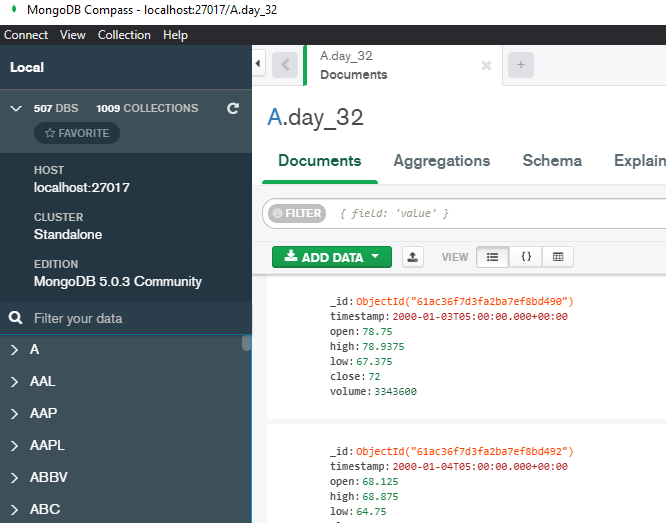
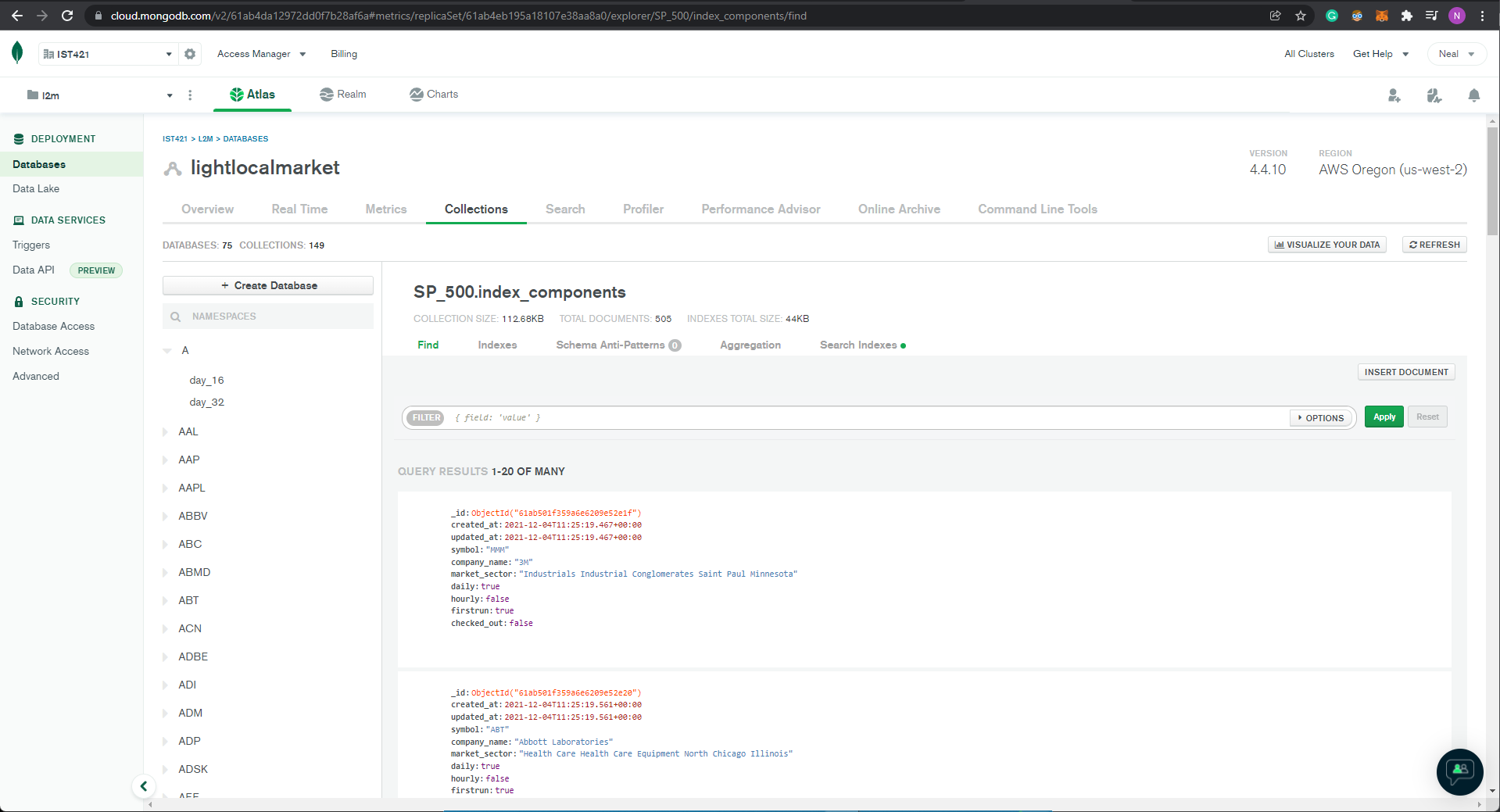


**Setup Instructions:**

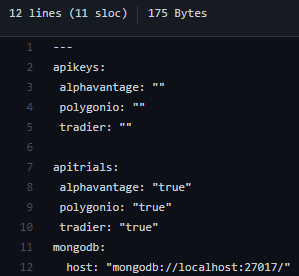
* Required:
  + Linux or Windows computer
  + Internet Connection
  + MongoDB
    - Local - <https://docs.mongodb.com/manual/administration/install-community/>
    - Cloud (Atlas) - <https://www.mongodb.com/cloud/atlas/>
    - Optional: Compass - <https://www.mongodb.com/products/compass/>
  + Free Developer API Access
    - Alpha Vantage - [https://www.alphavantage.co/support/#api-key](https://www.alphavantage.co/support/" \l "api-key)
    - Tradier - <https://tradier.com/solutions/developers>
  + lightlocalmarket binaries - <https://github.com/plihelix/l2m/>

MongoDB can be either local or cloud. The cloud service, called ‘Atlas’, is limited in free use and will not fully populate without upgrading into a paid service tier. Though this is a nominal fee, cloud use is better associated with a group of people working off of one installation further marginalizing the cost.

Pictured below is the data deployed in Atlas (left) and local viewed through Compass(right).



Once the appropriate installations are complete, the configuration file “.l2m” needs to be updated with the API keys and MongoDB location. The configuration file is a simple YAML text file and can be opened in your favorite Notepad utility for editing.



The API keys are alphanumeric strings that authenticate the origination of data requests. Each identifies you to the API and allows traffic to be throttled. If you share this key with others, their use will simply cost your available requests.

Currently the “apitrials:” values are not functional and the program assumes a 20 second delay between requests. Please insert the Developer API key retrieved from Alpha Vantage and Tradier into the quotes without additional spaces.

The “mongodb” host format can include login in the following format:

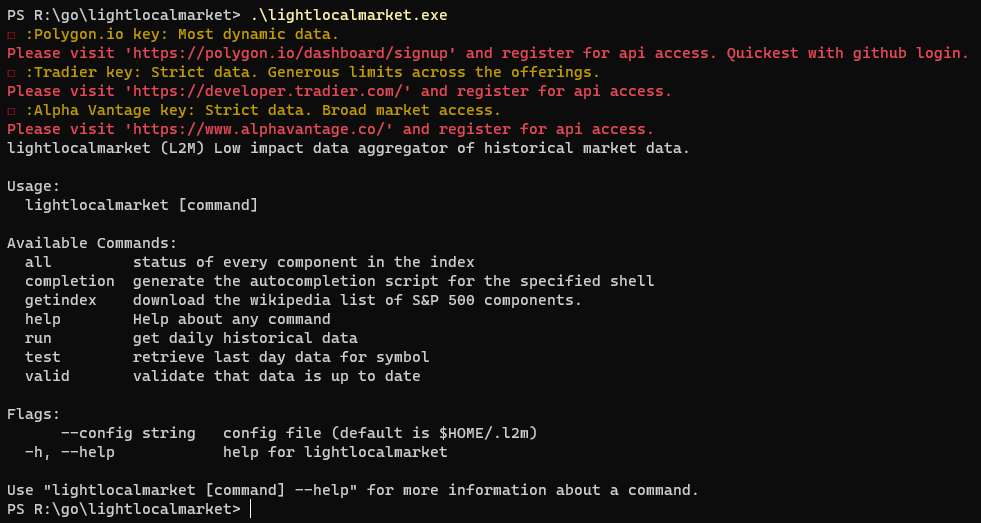
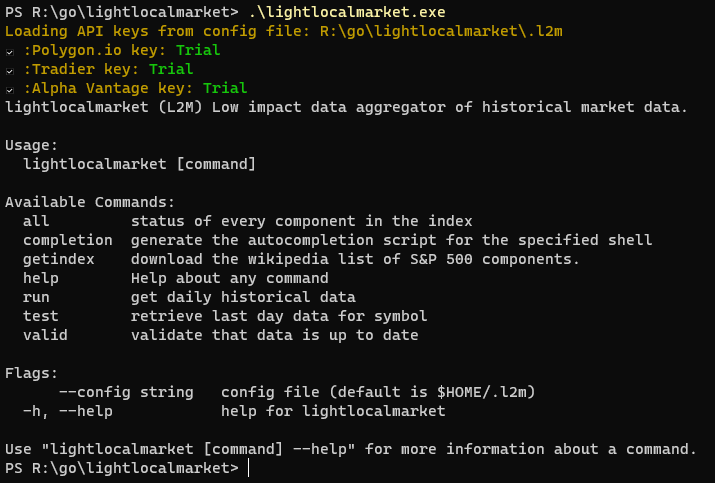
mongodb+srv://<username>:<password>@<cluster.address>.mongodb.net

/myFirstDatabase?retryWrites=true&w=majority

This means that the local login for a user “Admin” and password “Pass” would be:

mongodb://Admin:[Pass@localhost](mailto:Pass@localhost):27017/

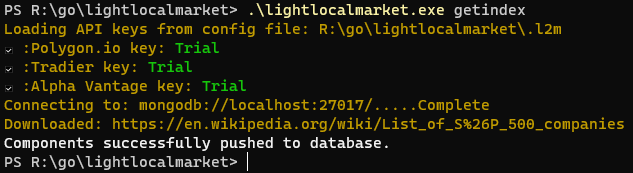
Running ‘lightlocalmarket.exe’ in Windows terminal or in ‘lightlocalmarket’ in a Linux terminal window will display the possible commands to run the integration.



If the configuration file is missing keys, the program will let you know.

**The Process to Fill and Update**

Before any historical data can be retrieved, the program needs to know what stocks it is to monitor. This is achieved through “Scraping” the Wikipedia page for the S&P 500 for the appropriate data, reformatting it, and injecting it into the database. The command to do this is ‘getindex’.

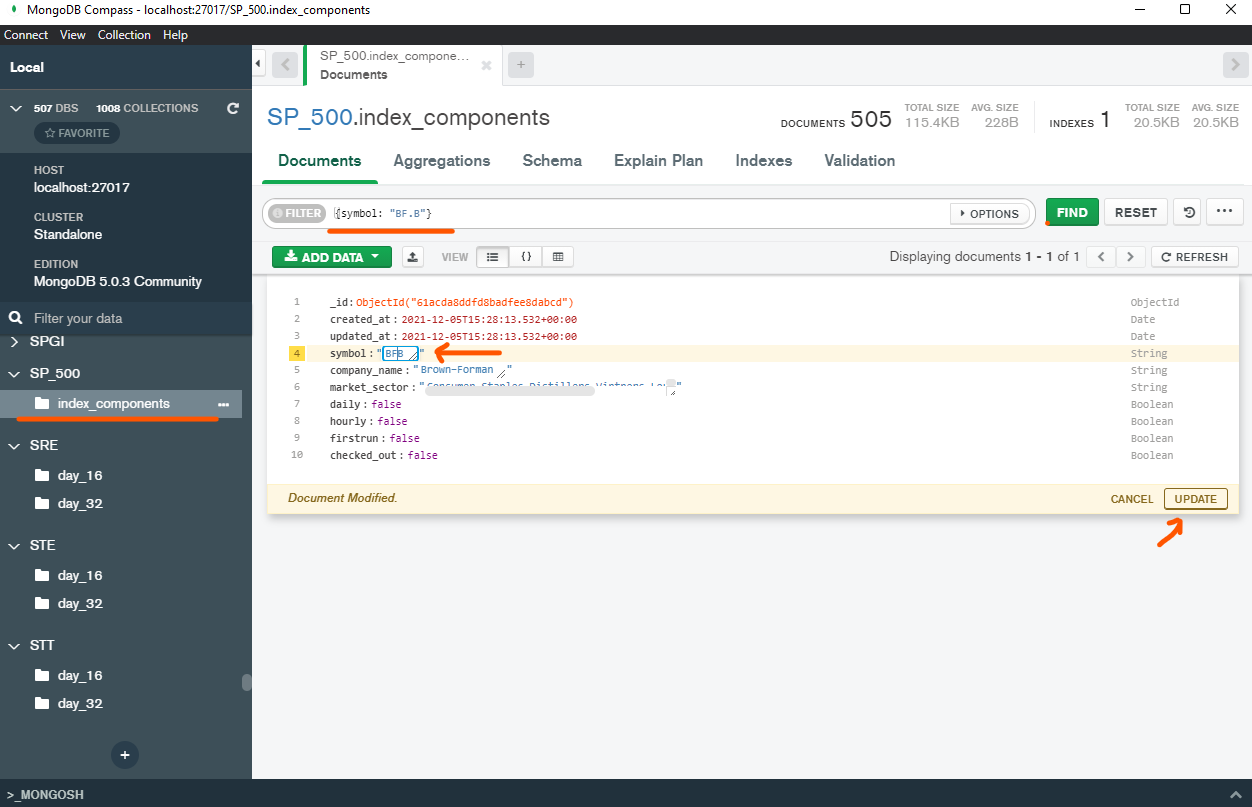


**Some changes must be made to the index in order to solve later problems:**

There are two stock symbols that have ‘.’ in them, MongoDB cannot include a period in the name of a database as it signals a distinction between the name and the collection within. As a result, the symbol values of BF.B and BRK.B need to be changed to BFB and BRKB respectively.

In Atlas or Compass, while connected to the same host, run the following search and modification.

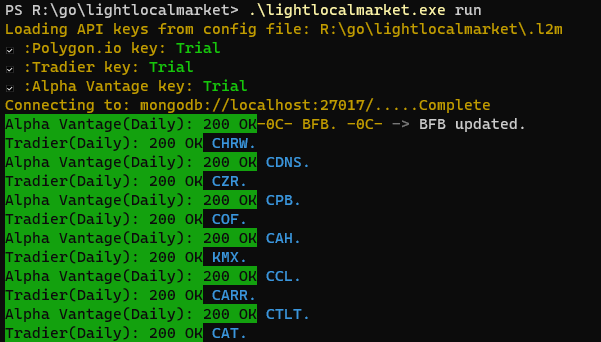
{symbol: “BF.B} as well as “BRK.B”}



With this workaround complete and those two entries updated, we can now check that the index has loaded properly with the command to display the status of ‘all’ entries.



To begin to fill the database, the command ‘run’ will compile a list of those without a green highlight on the “D” for daily. It will task this list in halves to Alpha Vantage and Tradier. The program checks out the symbol, requests the data, creates or updates the database for that symbol, and checks it back into the SP\_500 index. As a result, this process can be interrupted and will resume when run again.



As seen above this process is verbose, but minimally. The request status returns the HTTP response of the request. If the symbol is missing a database, the candle checks fail resulting in the “-0C-” code, and the program uses the appropriate command to insert the new data. If the candle checks don’t fail because there is already some data, a different command is used to ensure that data is not being duplicated, resulting in the blue completion text of the symbol.

As this process is complete, the ‘all’ command will show that the daily data is valid. When it is time to update the database, the ‘valid’ command will check that the last data entry for each symbol is not more than one day old. As a result, running ‘valid’ will update the SP\_500 index components so that the next ‘run’ will update with the most recent data.

The ‘valid’ command involves 1010 sort functions, and touches every single one of the approximately 3.6 million documents in the collections. Performance of this function will greatly depend on the cloud or computer capabilities.

Daily update sequence: (Windows)

1. Run: lightlocalmarket.exe valid
2. Run: lightlocalmarket.eve run