

This is going to be used as a place for me to dump all my thought processes during the completion of the test. First off, I lost a lot—and I mean A LOT—of time trying to figure out why the series IDs that I was building weren't working. At some point, I came back to the API docs and searched one of the examples that were given and realized that the first character could be omitted. From that point, I felt that everything was possible.

The first question code is located in the `cpi.py` file. I simply used the `requests` library to make a request post to the API with my registration key in a `.env` file, as it should be. When I received data back, I started getting only the fields that I was going to need. To gather all three prices in a single date, I used a `defaultdict` with an empty list as a default value just for simplicity. All that was left was to put the dict data into the CSV file.

The second question needs no further explanation; the code is located in the `chart.py` file. First, I read the CSV file using the `pandas` library, and then, I plot a line for all items, except food and energy, as was asked. Then, I save the output file inside the `out` folder, named `chart.html`.

The third and fourth questions are located in the `automation` and `correlation` files, respectively.

For the implementation of the fourth question, I simply did the same as I did in the second question; the only extra work was to plot a second line in the y-axis. The output file is also in the `out` folder, named `correlation.html`.

For the final bonus question, I created three distinct routes for each item, so they can be accessed individually. First, I read the CSV file, drop the columns that are not necessary for that specific item, rename to standardize the return data, and parse to a Python dict. While I was doing it, I also thought about implementing query parameters in the future for specific dates, prices, and so on.