Code Challenge - Exchange Rate

The Octane11 team is interested in having a tool for easily collecting historical exchange rates between multiple currencies as well as doing exchange rate conversions for specific values and dates.

We would like to write a Command Line Tool (CLI) using the language of your choice (Python, Go or JavaScript are preferred if you are choosing between languages) for satisfying the needs with the requirements below.

So, set your own pace. No rush. Just so you have some bounds, the challenge was designed to take somewhere between 3 and 4 hours. Set up your own repo and send us the link when ready. Have fun!

The solution should be properly **documented**, include **testing** and be **deployable**.

Requirements

The CLI has a command named `history` which pulls historical exchange conversions in a date range for a base currency and multiple other currencies. Arguments:

- `start` - the start date, default today.
- `end` - the end date, default today.
- `base` - the base currency, default USD.
- `symbol` - the list of currency symbols to convert to (space separated), required.
- `output` - the file name to write the output to, optional.

It dumps the output in the standard output (or in a file if the argument `output` is passed) in JSONlines. An example:

```
>$./exrates history --start 2021-02-01 --end 2021-02-02 --base USD --symbol EUR
CAD
{"date":"2021-02-01", "base":"USD", "symbol":"CAD", "rate":1.2805362463}
{"date":"2021-02-01", "base":"USD", "symbol":"EUR", "rate":0.8275405495}
{"date":"2021-02-02", "base":"USD", "symbol":"CAD", "rate":1.2804716041}
{"date":"2021-02-02", "base":"USD", "symbol":"EUR", "rate":0.8302889406}
```

You can use the free API provided by Frankfurter for getting the data.

Optional

If you are inspired and want to go further, you can add another command called `convert` for doing currency conversion. Arguments:

- `date` - the currency exchange date, default today.
- `base` - the base currency, default USD.
- `symbol` - the currency symbol to convert to, required.
- `amount` - the amount to convert, required.

```
>$./exrates convert --date 2021-02-01 --base USD --symbol EUR --amount 50
41.377027475
```