Candidate Project: Currency Conversion

Thank you for taking the time to apply to our candidate position, and for agreeing to this programming challenge. The intent here is to get a feel for current coding abilities and styles.

The end goal for this project is to create a command-line program/script that allows a user to do monetary conversions from one currency to one or more currencies.

Inputs:

- Starting Currency Type
- Starting Currency Amount
- Desired Target Currencies (1 or more)

Output:

- The output must include (but is not limited to):
- The given inputs
- The date the conversion was run
- Each output conversion type, rate, amount, and the time the rate last changed.

It is acceptable to use a default input and output currency type(s), or other assumptions. However, all assumptions must be well documented. Usage of 3rd party modules is optional, but encouraged.

Goal 1: Installation

- A user must be able to clone the project from a public github repository to any machine and run the program.
- Any details in regards to the installation process must be documented and accurate.
- Prerequisites must be explicitly stated, including any limitations to program versions or operating systems, etc.

Goal 2: Running program with help

- The program can run as a server or as a single-execution script.
- The program must be written in JavaScript using node.js. TypeScript is also an option.
- The program should have a method to get detailed help.
- Please also include and/or document your approach to testing the scenario.

Goal 3: Conversion

- Conversion rates must use real-time data.

- The format of input arguments are up to the developer, however they must be clearly stated.

Goal 4: Database - Historical lookup

- The user must be able to retrieve a history of conversions performed.
- It is expected that there will be/can be a maximum number of historical conversions returned.
- MongoDB is preferred
- History remains when script ends.
- Schema is clearly stated.

Goal 5: Additional Output Formats

Note: This is viewed as a ‘nice to have’.

The user will have the ability to save the full database as historical records to one or more outputs: csv, xlsx, pdf, txt.   The user can specify the name and location of the file.

Goal 6: Formal Test Suite

Note: This is viewed as a ‘nice to have’.

- Enable a test suite for the program.
- Known inputs should give known output.
- Include expected passing cases.
- Include expected failure cases.