Take-Home Assignment

We are expanding into the algorithmic trading of tropical fruits!

We need your help to build a REST API called **fruittrade,** which allows a trader to understand the full cost of buying fruit from various countries of origin.

Inputs

A trader using this API will specify the contemplated purchase, including:

1. The commodity (e.g. mangos)
2. The price per ton (in dollars)
3. The trade volume (in tons)

Example

A trader who wants to know the full cost of buying 405 tons of mangos at $53 a ton would call the calculate endpoint with the following parameters: `COMMODITY=mango PRICE=53 TONS=405`

Outputs

The API must return a JSON array of all available countries of origin, and, for each:

- The country code
- The total cost of the purchase
- The breakdown of the costs between variable and trade cost, as shown in the example

The list must be sorted by total cost from highest to lowest.

Example

In response to the Example Input of `COMMODITY=mango PRICE=53 TONS=405`, the API must return:

```
[{
        "COUNTRY":"BR",
        "TOTAL_COST":"22060.10",
        "TRADE_COST":"20.00",
        "VARIABLE_COST":"54.42"
},{
        "COUNTRY":"MX",
        "TOTAL_COST":"21999.20",
        "TRADE_COST":"32.00",
        "VARIABLE_COST":"54.24"
}]
```

This means the 405 tons of mangoes would cost $22,060.10 if bought in Brazil and $21,999.20 if bought in Mexico. The Brazilian mangoes cost $54.42 per-ton and $20 per-trade.

Calculations

The cost is computed from a combination of the trader’s inputs with market trade data (fees and taxes) that we are getting from a 3rd-party API.

- The cost per ton is calculated as the price from the trader + the “variable cost” from our data vendor.

3rd-Party Data

We have a deal with a 3rd-party that provides us the latest trade cost and variable cost data for various fruits across a variety of countries.

The data is provided to us on a nightly basis as a flat file transmitted via FTP (you can assume that your server will have access to the file locally).

The file contains JSON with the following format (data will be different):

```
[{
        "COUNTRY":"MX",
        "COMMODITY":"mango",
        "TRADE_COST": "32.00",
        "VARIABLE_COST":"54.24"
},
{
        "COUNTRY":"BR",
        "COMMODITY":"mango",
        "TRADE_COST":"20.00",
        "VARIABLE_COST":"54.42"
},
{
        "COUNTRY":"PA",
        "COMMODITY":"mango",
        "TRADE_COST":"20.00",
        "VARIABLE_COST":"44.42"
},
{
        "COUNTRY":"PA",
        "COMMODITY":"raspberry",
        "TRADE_COST":"20.00",
        "VARIABLE_COST":"34.12"
}]
```

Considerations

- Assume that this is a proof of concept that will eventually be plugged into a larger software system. You should treat this like a piece of core business logic rather than a throwaway script.
- Implement your solution as production quality code that will be maintained by a team of engineers in the long run. You or your teammates should be able to come back to this code in six months and be able to easily extend it.
- The stability of your system is important, so consider writing some automated tests or using some other best practices to ensure it is stable and correct, such as creating additional test input.
- You are going to have onsite interview in which we are going to ask to create a web-based frontend to use this API.