# Performativ Tech Challenge

The objective of this coding challenge is to have you implement a calculator for a subset of simplified financial metrics.

You can address this challenge by using any language of your preference.

Before starting make sure you have been granted access to the tech-challenge API and that you have a functioning API Key.

You should start by creating a private GitHub repository and granting access to the tech-challenge-reviewer user.

You can structure your project and code as you see best fit to achieve the correct calculation of the metrics listed on page 3.

The steps to complete the challenge are:

- Create a method that takes the set of **positions**, the target currency (TC) **USD**, and the time window **2023-01-01** to **2024-11-10** as its inputs.
- The method should leverage the API endpoints at your disposal.
- Follow the mathematical definitions provided on page 3 to guide you through the implementation of all the metrics requested.
- The method should return the complete result, not a partial one, and be easily extendable to other metrics.
- Use the submit endpoint to upload the result to this challenge. There is no limit to the number of times you can submit a result - only the one you share with us will be used to review your candidacy.
- Only the following metrics are to be submitted: **IsOpen, Price, Value, ReturnPerPeriod, and ReturnPerPeriodPercentage.**

**Submission Details**

To complete this challenge you will need to upload your result to the submit endpoint. The endpoint will return you a unique ID as well as a percentage score that can help you evaluate your submission. **Once done, if happy with the result, we ask you to screenshot the response from your request and share it along with the ID with your Performativ HR reference.**

**Financial Metrics**

Financial metric values, for the scope of this challenge, can be either monetary or percentage values. Position monetary metric values are additive when accounted in the same target currency, while percentage values are not.

In this challenge, we ask you to compute metric values in two timeseries:

- At the positional level. Each metric listed above should have a value for each position on each day of the time window.
- At the basket level. Each metric should have a value for the aggregate of all positions on each day of the time window.

**Positions**

Positions are the driving input behind any valuation request, representing the user's holdings within a specific time window.

A position describes the holding amount of a specific instrument in a portfolio. The summation of all position monetary metric values is the basket value of the respective metric being calculated.

```
id: 24141
open_date: "2021-01-15"
close_date: "2022-02-15"
open_price: "152.21451251"
close_price: "98.124214"
quantity: 142.21
transaction_costs: 0.0
instrument_id: 413
instrument_currency: GBP
open_transaction_type: "BUY"
close_transaction_type: "SELL"
```

The `open_date` and `close_date` are the dates the position opened and closed.

The `open_price` and `close_price` are the prices at which the instrument was bought and sold.

The `quantity` is how many shares of an instrument the position represents.

The `instrument_currency` is the currency of the instrument prices, referred to as local currency (LC).

*Note: Transaction costs and intraday forex rates have been excluded from the metrics definitions to simplify calculations.*

### General Guidelines

Only query data within the scope of the dates and positions provided. If the API does not return a result for your query you will not need it to complete the implementation.

Assume the value for $\text{FxRate}_{TC \to TC,t} = 1$. The API will not return this conversion.

The result submitted in the body of your request should have the following shape:

```
{
  positions: {
    299825: {
      IsOpen: [1.0, 1.0, … ],
      Price: [ … ],
      Value: [ … ],
      ReturnPerPeriod: [ … ],
      ReturnPerPeriodPercentage: [ … ],
    },
    …
    299845: { … },
  },
  basket: {
    IsOpen: [1.0, 1.0, … ],
    Price: [ … ],
    Value: [ … ],
    ReturnPerPeriod: [ … ],
    ReturnPerPeriodPercentage: [ … ],
  },
  dates: [`2023-01-01`, … ]
}
```

Numerical results are evaluated with eight decimal points precision.

The financial metrics arrays should be of length 680, one value for each day in the range between start and end date.

Ordering of the fields is not important.

### API Specs

api.challenges.performativ.com

To complete this challenge, you have three endpoints at your disposal. To query the API add the `x-api-key` header to your requests, setting the value to the provided API Key:

```
– header: 'x-api-key: 94bM7T5kn39y5c3oweiV76GJf…'
– header: 'Accept: application/json'
```

**GET** **/fx-rates**

Query Params

| Parameter | Type | Required | Description | Example |
|---|---|---|---|---|
| `pairs` | string | Yes | Comma separated string of currency pairs. A pair is composed by local followed by target currency. | USDGBP,GBPDKK |
| `start_date` | string | Yes | Start date formatted as YYYYMMDD | 20241116 |
| `end_date` | string | Yes | End date formatted as YYYYMMDD | 20241116 |

**GET** **/prices**

Query Params

| Parameter | Type | Required | Description | Example |
|---|---|---|---|---|
| `instrument_id` | String | Yes | Instrument id | 11 |
| `start_date` | string | Yes | Start date formatted as YYYYMMDD | 20241116 |
| `end_date` | string | Yes | End date formatted as YYYYMMDD | 20241116 |

**POST** **/submit**

Body

| Parameter | Type | Required | Description | Example |
|---|---|---|---|---|
| `positions` | json | Yes | The positional result. | |
| `basket` | json | Yes | The basket result. | |

## Metrics to implement:

$i$ indexes positions, $t$ indexes days.

**OpenPriceLocal**

The position's `open_price` listed on its DTO.

$$\text{OpenPrice}_{i,t}^{LC} = \text{The position's open price}$$

$$\text{OpenPrice}_{t}^{LC} = 0$$

**OpenPrice**

Calculated using the position's converted `open_price`.

$$\text{OpenPrice}_{i,t}^{TC} = \text{OpenPrice}_{i,t}^{LC} \times \text{FxRate}_{LC \to TC, Open}$$

$$\text{OpenPrice}_{t}^{TC} = 0$$

where:

$\text{FxRate}_{LC \to TC, Open}$ = The forex rate from local to target currency on the open date of the position.

**ClosePriceLocal**

The position's `close_price` listed on its DTO.

$$\text{ClosePrice}_{i,t}^{LC} = \text{The position's close price}$$

$$\text{ClosePrice}_{t}^{LC} = 0$$

**ClosePrice**

Calculated using the position's converted `close_price` from instrument currency to target currency.

$$\text{ClosePrice}_{i,t}^{TC} = \text{ClosePrice}_{i,t}^{LC} \times \text{FxRate}_{LC \to TC, Close}$$

$$\text{ClosePrice}_{t}^{TC} = 0$$

where:

$\text{FxRate}_{LC \to TC, Close}$ = The forex rate from local to target currency on the close date of the position.

**PriceLocal**

The market price of the position's instrument returned in its local currency.

$$\text{Price}_{i,t}^{LC} = \begin{cases} \text{Instrument price at time } t, \text{ if } t \geq \text{open\_date} \\ 0, \text{ if } t < \text{open\_date} \end{cases}$$

$$\text{Price}_{t}^{LC} = 0$$

**Price**

The market price of the position's instrument returned in the valuation request's target currency.

$$\text{Price}_{i,t}^{TC} = \text{Price}_{i,t}^{LC} \times \text{FxRate}_{LC \to TC, t}$$

$$\text{Price}_{t}^{TC} = 0$$

where $\text{FxRate}_{LC \to TC, t}$ = The forex rate from local to target currency at time *t*.

**All metrics should be implemented at the positional (indexed over i,t) and basket level (indexed over t), resulting in a response object that contains exactly two collections of timeseries. We assume *t* to iterate from start date to end date of the request with both included.**

**Quantity**

The `quantity` of each position in the valuation request `positions` set.

$$\text{Quantity}_{i,t} = \begin{cases} \text{Position quantity, if open\_date} \leq t < \text{close\_date} \\ 0, \text{ otherwise} \end{cases}$$

$$\text{Quantity}_{t} = \sum_{i=1}^{N} \text{Quantity}_{i,t}$$

**IsOpen**

The open status of the position, dictated by its `open_date` and `close_date` properties.

$$\text{IsOpen}_{i,t} = \begin{cases} 1, \text{ for open\_date} \leq t < \text{close\_date} \\ 0, \text{ otherwise} \end{cases}$$

$$\text{IsOpen}_{t} = \bigvee_{i=1}^{n} \text{IsOpen}_{i,t} \text{ (bitwise)}$$

where position i is open at time t when `open_date` <= t < `close_date` and closed otherwise.

**ValueLocal**

The value of a position in the instrument's local currency.

$$\text{Value}_{i,t}^{LC} = \text{Price}_{i,t}^{LC} \times \text{Quantity}_{i,t}$$

$$\text{Value}_{t}^{LC} = 0$$

**Value**

The value of a position and the basket of positions in the valuation target currency.

$$\text{Value}_{i,t}^{TC} = \text{Value}_{i,t}^{LC} \times \text{FxRate}_{LC \to TC, t}$$

$$\text{Value}_{t}^{TC} = \sum_{i=1}^{n} \text{Value}_{i,t}^{TC}$$

where:

$\text{FxRate}_{LC \to TC, t}$ = The forex rate from local to target currency at time *t*.

**OpenValue**

The value of a position according to the `open_price` listed in its DTO.

$$\text{OpenValue}_{i,t}^{TC} = \begin{cases} \text{OpenPrice}_{i,t}^{TC} \times \text{Quantity}_{i,open\_date}, \text{ if } t \geq \text{open\_date} \\ 0, \text{ if } t < \text{open\_date} \end{cases}$$

$$\text{OpenValue}_{t}^{TC} = \sum_{i=1}^{n} \text{OpenValue}_{i,t}^{TC}$$

**CloseValue**

The value of a position according to the `close_price` listed in its DTO.

$$\text{CloseValue}_{i,t}^{TC} = \begin{cases} \text{ClosePrice}_{i,t}^{TC} \times \text{Quantity}_{i,open\_date}, \text{ if } t \geq \text{close\_date} \\ 0, \text{ if } t < \text{close\_date} \end{cases}$$

$$\text{CloseValue}_{t}^{TC} = \sum_{i=1}^{n} \text{CloseValue}_{i,t}^{TC}$$

**ReturnPerPeriod**

The daily monetary value return generated by the basket of positions.

$$\text{RPP}_{i,t} = \begin{cases} \text{Value}_{i,End}^{TC} - \text{Value}_{i,Start}^{TC}, \text{ if open\_date} \leq t \leq \text{close\_date} \\ 0, \text{ otherwise} \end{cases}$$

$$\text{Value}_{i,Start}^{TC} = \begin{cases} \text{Value}_{i,t}^{TC}, & \text{if } t \neq \text{open\_date} \wedge t = \text{start\_date} \\ \text{OpenValue}_{i,t}^{TC}, & \text{if } t = \text{open\_date} \\ \text{Value}_{i,t-1}^{TC}, & \text{if } t > \text{open\_date} \wedge t > \text{start\_date} \end{cases}$$

$$\text{Value}_{i,End}^{TC} = \begin{cases} \text{Value}_{i,t}^{TC}, & \text{if } t < \text{close\_date} \\ \text{CloseValue}_{i,t}^{TC}, & \text{if } t = \text{close\_date} \end{cases}$$

$$\text{RPP}_{t} = \sum_{i=1}^{n} \text{RPP}_{i,t}$$

**ReturnPerPeriodPercentage**

The percentage daily return generated by the basket of positions.

$$\text{RPP\%}_{i,t} = \begin{cases} \dfrac{\text{RPP}_{i,t}}{\text{Value}_{i,Start}^{TC}}, \text{ for Value}_{i,Start}^{TC} \neq 0 \\ 0, \text{ otherwise} \end{cases}$$

$$\text{RPP\%}_{t} = \begin{cases} \dfrac{\sum_{i=1}^{n} \text{RPP}_{i,t}}{\sum_{i=1}^{n} \text{Value}_{i,Start}^{TC}}, \text{ for } \sum_{i=1}^{n} \text{Value}_{i,Start}^{TC} \neq 0 \\ 0, \text{ otherwise} \end{cases}$$