Python Assignment Instruction Manual

The modules are designed to be run as executables in Windows/Linux environments, please follow the below steps to execute the modules and obtain the outputs for review.

1. Copy the executables on to a system folder in your virtual machine
2. Copy the attached input files (firm_trades.csv and task2_stock_data.xlsx) onto a folder which location can be passed as parameters to run the executables

Task 1:

Please use the following command to run the executable for REST API from the location where you have executable,

For windows:

```
start start_server_winx64.exe --csv-file <path for csv> --db-file <path for DB>
```

For Linux:

```
start_server_linux --csv-file /absolute-path/to/csv-file --db-file C:\path\to\db-file
```

For example:

```
start start_server_winx64.exe --csv-file C:\path\to\csv-file --db-file C:\path\to\db-file
```

This should kickstart the webserver with Rest API which can provide the required JSON outputs for the listed 3 different parameters, you can use the browser or curl commands to obtain the output for analysis.

Scenario 1: Parent Order Object and Underlying fill data. Given: order_id Return: top level order object with grouped fill objects.

```
Example:
curl -X GET -H "Content-Type: application/json" "http://localhost:4444/orders/get/3520680"
```

Scenario 2: Additional query options to search orders by ticker, and by date range.

Example:

```
curl -X GET -H "Content-Type: application/json"
"http://localhost:4444/orders/search?date_from=2022-06-01&date_to=2022-06-30&ticker=FNOE"
```

Scenario 3: Returning Order summary statistics, given all the above defined search parameters (order_id, ticker, and date range).

Example:

```
curl -X GET -H "Content-Type: application/json"
"http://localhost:4444/orders/summary?date_from=2022-06-01&date_to=2022-06-
30&order_id=3520680&ticker=FNOE"
```

Task 2:

Please use the following command to run the executable for beta calculation from the location where you have executable,

For windows:

```
start beta_calc_win-x64.exe --excel-file <path for xslx> --as-of-date AS_OF_DATE [--window 1y
WINDOW] --frequency {daily, weekly, monthly, quarterly, bi-weekly}
```

For Linux:

```
beta_calc_linux --excel-file /absolute-path/to/xlsx-file --as-of-date AS_OF_DATE [--window
WINDOW] --frequency {daily,weekly,monthly,quarterly,bi-weekly}
```

For example:

```
start beta_calc_win-x64.exe --excel-file C:\absolute-path\to\xlsx-file --as-of-date AS_OF_DATE
--window 1y --frequency daily
```

```
start beta_calc_win-x64.exe --excel-file C:\absolute-path\to\xlsx-file --as-of-date AS_OF_DATE
--window 1y --frequency monthly
```

This should run the calculation with requested conditions and produce the JSON output onto the same folder where the executables were located.