Development Challenge

Power trade position (C#)

Overview

Power traders need an intra-day report that provides their consolidated day-ahead power position, in other words, it is a report of the forecast of the total energy volume per hour required by Axpo for the next day. The report should generate an hourly aggregated volume and save it to a CSV file, following a schedule that can be configured.

Requirements

1. Must be implemented as a console application using .Net Core 8 (or higher) using either C# or F#.
2. The CSV:
    a. It has two columns `Datetime` and `Volume`.
    b. The first row is the header.
    c. Semi-column is the separator.
    d. The point is the decimal separator.
    e. All trade positions shall be aggregated per hour (local/wall clock time).
    f. To avoid any misunderstanding with dates or time zones, the column `Datetime` should be in UTC.
    g. The `Datetime` format should follow ISO_8601.
    h. You should mind any potential issue with Daylight Saving Time.
3. CSV filename must follow the convention *PowerPosition_YYYYMMDD_YYYYMMDDHHMM.csv* where YYYYMMDD is year/month/day, e.g. 20141220 for 20 Dec 2014 and HHMM is 24hr time hour and minutes e.g. 1837. The first date refers to the day of the volumes (the day-ahead) while the second datetime refers to the timestamp of the extraction in UTC.
4. The folder path for storing the CSV file can be either supplied on the command line or read from a configuration file.
5. An extract must run at a scheduled time interval; every X minutes where the actual interval X is passed on the command line or stored in a configuration file. This extract does not have to run exactly on the minute and can be within +/- 1 minute of the configured interval.
6. It is not acceptable to miss a scheduled extract, therefore a retry mechanism should be in place.
7. An extract must run when the application first starts and then run at the interval specified as above.
8. The application must provide adequate logging for production support to diagnose any issues.

Additional Notes

1. An assembly (.net standard 2.0) has been provided (PowerService.dll) that must be used to interface with the “trading system”.
2. A single interface is provided to retrieve power trades for a specified date. Two methods are provided, one is a synchronous implementation (`IEnumerable<PowerTrade> GetTrades(DateTime date);`) and the other is asynchronous (`Task<IEnumerable<PowerTrade>> GetTradesAsync(DateTime date);`). The implementation can use either of these methods. The class `PowerService` is the actual implementation of this service.
3. The argument `date` refers to the reference date of the trades thus, you will need to request the date of the following day if you want to get the power positions of the day-ahead. Which is an array of `PowerTrade's`.

4. The `PowerTrade` class contains an array of `PowerPeriods` for the given day. The period number follow a numeric sequency starting at 1, which corresponds to the first period of the day.
5. The service `PowerService` intends to provide trades for a specific country/location, but the return of `PowerPeriods` don't have a **specified time zone**, although it internally considers a specifically the location, Europe\Berlin. Your application should consider this location and convert to UTC for the output. We shouldn't trust the server settings and have the location configured in the application. Therefore, the resultant CSV should be the same independently of the location of the server and its configuration.
6. The completed solution must include all source code and be able to be compiled from source. It may be delivered as (in order of preference) a cloud storage link to a zip file, a link to a hosted source control repository, or as a zipped email attached. If you send a zipped attachment via email, please do not include the actual compiled executable in the zip and send a separate email that states that you have sent your solution via email.

Example

On the 2023-07-01, the application is triggered and a call to `PowerService.GetTrades` requesting data for the day-ahead, 2023-07-02, returns a list with the following two trade positions:

Date	2023-07-02
Period (#)	Volume
1	100
2	100
3	100
4	100
5	100
6	100
7	100
8	100
9	100
10	100
11	100
12	100
13	100
14	100
15	100
16	100
17	100
18	100
19	100
20	100
21	100

Date	2023-07-02
Period (#)	Volume
1	50
2	50
3	50
4	50
5	50
6	50
7	50
8	50
9	50
10	50
11	50
12	-20
13	-20
14	-20
15	-20
16	-20
17	-20
18	-20
19	-20
20	-20
21	-20

22	100
23	100
24	**100**

22	-20
23	-20
24	**-20**

Considering you asked data at `2023-07-01T21:15:00 (Europe/Madrid)`, the expected output would be a CSV named `PowerPosition_20230702_202307011915.csv` containing the following data:

Datetime	Volume
2023-07-01T22:00:00Z	150
2023-07-01T23:00:00Z	150
2023-07-02T00:00:00Z	150
2023-07-02T01:00:00Z	150
2023-07-02T02:00:00Z	150
2023-07-02T03:00:00Z	150
2023-07-02T04:00:00Z	150
2023-07-02T05:00:00Z	150
2023-07-02T06:00:00Z	150
2023-07-02T07:00:00Z	150
2023-07-02T08:00:00Z	150
2023-07-02T09:00:00Z	80
2023-07-02T10:00:00Z	80
2023-07-02T11:00:00Z	80
2023-07-02T12:00:00Z	80
2023-07-02T13:00:00Z	80
2023-07-02T14:00:00Z	80
2023-07-02T15:00:00Z	80
2023-07-02T16:00:00Z	80
2023-07-02T17:00:00Z	80
2023-07-02T18:00:00Z	80
2023-07-02T19:00:00Z	80
2023-07-02T20:00:00Z	80
2023-07-02T21:00:00Z	80