# Exercise Leakzon

1. Write an API for consumptions with .NET Core 6+.
2. The API should run in a docker container and have connection to a DB of your choice that also runs in a container.
3. Write a docker-compose for it.
4. Check the attached files in the zip folders.
5. The API should contain the following endpoints.
6. Post readings to the DB.
   1. Input is a list of readings (in zip file 'rawdata.json')
7. Get Latest reading (by date) of a sensor or for all existing sensors.
   1. Input: Sensor Id
      1. If the sensor Id is null, it returns last reading for every sensor in list.
      2. Else it returns the latest reading for the specified sensor Id.
   2. Output: List of Readings
8. Get Oldest reading of a sensor (by date) or for all existing sensors.
   1. Input: Sensor Id.
      1. If the sensor Id is null, it returns last reading for every sensor in list.
      2. Else it returns the oldest reading for the specified sensor Id.
   2. Output: List of Readings.
9. Get consumption for a sensor or for all existing sensors.
   1. Input: **Range of dates** and **Sensor Id list**. (from, to, sensorIds [])
   2. The endpoint **calculates (**explanation below) the hourly consumption from the readings in the DB for the specified sensor Id.

If the Sensor Id list is null, then calculate the consumption for every sensor in the DB that matching the date range.

* 1. Output: csv file of **hourly** consumptions for the sensors. (example table B)

(if table A is the DB, then table B is the output)

Consumption

The consumption is the difference between 2 readings of the **same sensor**.

We can rephrase it like this. The difference between 2 readings is the consumption between their dates, we are writing the consumption on the last reading time.

Pay attention to the case that you have more then 2 reading in one hour.

Example Table A readings

|  |  |  |  |
| --- | --- | --- | --- |
| Id | SensorId | ReadingValue | Date |
| 121045832 | 1300001102 | 39.99 | 05/02/2024 16:01 |
| 120873254 | 1300001102 | 39.79 | 05/02/2024 15:31 |
| 121045833 | 1300001102 | 39.77 | 05/02/2024 15:08 |
| 120873255 | 1300001102 | 39.77 | 05/02/2024 14:58 |
| 121045834 | 1300001102 | 39.76 | 05/02/2024 14:08 |
| 120873256 | 1300001102 | 39.75 | 05/02/2024 13:48 |
| 121045835 | 1300001102 | 39.74 | 05/02/2024 13:31 |
| 120873257 | 1300001102 | 39.72 | 05/02/2024 12:08 |
| 121045836 | 1300001102 | 39.71 | 05/02/2024 12:08 |
| 120873258 | 1300001102 | 39.70 | 05/02/2024 11:28 |
| 121045837 | 1300001102 | 39.69 | 05/02/2024 11:08 |
| 120873259 | 1300001102 | 39.64 | 05/02/2024 10:18 |
| 121045838 | 1300001102 | 39.63 | 05/02/2024 10:08 |

Example Table B Consumptions

|  |  |  |  |
| --- | --- | --- | --- |
| SensorId | ReadingValue | Date | Consumption |
| 1300001102 | 39.99 | 05/02/2024 16:00 | 0.22 |
| 1300001102 | 39.77 | 05/02/2024 15:00 | 0.01 |
| 1300001102 | 39.76 | 05/02/2024 14:00 | 0.02 |
| 1300001102 | 39.74 | 05/02/2024 13:00 | 0.03 |
| 1300001102 | 39.71 | 05/02/2024 12:00 | 0.03 |
| 1300001102 | 39.69 | 05/02/2024 11:00 | 0.06 |