Opening Hours

Version 2.2

In short

Your task is to write an endpoint that accepts **JSON-formatted** opening hours of a restaurant as an input and returns the rendered **human readable format** as a text output.

Input data

Input JSON consists of keys indicating days of a week and corresponding opening hours as values. One JSON file includes data for one restaurant.

{
    <dayofweek>: <opening hours>
    <dayofweek>: <opening hours>
    ...
}

<dayofweek>: *monday* / *tuesday* / *wednesday* / *thursday* / *friday* / *saturday* / *sunday*
<opening hours>: an array of objects containing opening hours. Each object consist of two keys:

- **type**: *open* or *close*
- **value***:* opening / closing time as UNIX time (1.1.1970 as a date),
  e.g. *32400 = 9 AM*, *37800 = 10.30 AM,*
  max value is *86399 = 11.59:59 PM*

Example: on Mondays a restaurant is open from 9 AM to 8 PM

{

```
    "monday" : [
      {
        "type" : "open",
        "value" : 32400
      },
      {
        "type" : "close",
        "value" : 72000
      }
    ],
    ....
}
```

Special cases

- If a restaurant is closed the whole day, an array of opening hours is empty.
    - *“tuesday”: []* means a restaurant is closed on Tuesdays

- A restaurant can be opened and closed multiple times during the same day,
    - E.g. on Mondays from 9 AM - 11 AM and from 1 PM to 5 PM

- A restaurant might not be closed during the same day
    - A restaurant can be opened e.g. on a Friday evening and closed early Saturday morning. In that case *friday*-object includes only the opening time. Closing time is part of the *saturday*-object.
    - When printing opening hours which span between multiple days, closing time is always a part of the day when a restaurant was opened (e.g. Friday 8 PM - 1 AM)

```
{
    "friday" : [
      {
        "type" : "open",
        "value" : 64800
      }
    ],
    “saturday”: [
      {
        "type" : "close",
        "value" : 3600
      },
      {
        "type" : "open",
        "value" : 32400
      },
      {
        "type" : "close",
```

A restaurant is open:
Friday: 6 PM - 1 AM
Saturday: 9 AM -11 AM, 4 PM - 11 PM

```
            "value" : 39600
        },
        {
            "type" : "open",
            "value" : 57600
        },
        {
            "type" : "close",
            "value" : 82800
        }
    ]
}
```

Deliverable

Part 1

Build a HTTP API that accepts opening hours data as an input (JSON) and returns a more **human readable version of the data formatted using a 12-hour clock**.

Output example in 12-hour clock format:

Monday: 8 AM - 10 AM, 11 AM - 6 PM
Tuesday: Closed
Wednesday: 11 AM - 6 PM
Thursday: 11 AM - 6 PM
Friday: 11 AM - 9 PM
Saturday: 11 AM - 9 PM
Sunday: Closed

Your API can print the formatted version to the console or return it to the caller (of the API).

Part 2

Tell us what you think about the data format. Is the current JSON structure the best way to store that kind of data or can you come up with a better version? There are no right answers here 🙂. **Please write your thoughts to *readme.md.***

Sending the assignment

Bundle everything into a Zip archive and mail it to us. Remember that it is easier for us to review your task if we can test & run it.

A good check before sending your task is to unzip the Zip archive into a new folder and check that building and running the project works, using the steps you define in readme.md. Forgotten dependencies and instructions can sometimes happen even to the best of us

A word about the expected quality

We consider this exercise as "**PR review**". Our developers will check code and prepare comments & questions they want to go through with you during the interview.

Send us code you would be happy to review by yourself and discuss further.

What else happens during the tech interview?

In addition to going through the homework assignment, we typically have a system design exercise. The task is not about memorizing sorting algorithms, but instead figuring out an MVP solution to something that could be built by one of the engineering teams in Wolt. There are no right or wrong answers and the exercise is done with a very discussion-oriented approach - the same way we work in Wolt.

Technologies

Feel free to use any programming language you prefer (unless specifically asked to write the program using e.g. Python or Scala). 3rd party libraries and frameworks are also allowed, but consider carefully do you really need them.

Full JSON Example

Input

```
{
    "monday" : [],
    "tuesday" : [
       {
          "type" : "open",
          "value" : 36000
```

```
        },
        {
            "type" : "close",
            "value" : 64800
        }
    ],
    "wednesday" : [],
    "thursday" : [
        {
            "type" : "open",
            "value" : 37800
        },
        {
            "type" : "close",
            "value" : 64800
        }
    ],
    "friday" : [
        {
            "type" : "open",
            "value" : 36000
        }
    ],
    "saturday" : [
        {
            "type" : "close",
            "value" : 3600
        },
        {
            "type" : "open",
            "value" : 36000
        }
    ],
    "sunday" : [
        {
            "type" : "close",
            "value" : 3600
        },
        {
            "type" : "open",
            "value" : 43200
        },
        {
            "type" : "close",
            "value" : 75600
        }
    ]
}
```

Output

Monday: Closed
Tuesday: 10 AM - 6 PM
Wednesday: Closed
Thursday: 10:30 AM - 6 PM
Friday: 10 AM - 1 AM
Saturday: 10 AM - 1 AM
Sunday: 12 PM - 9 PM