Electronic Program Guide

Summary

Your task is to write a web service exposing an endpoint that accepts **JSON formatted** input of Electronic Program Guide (EPG) and returns a **human readable format** as text output.

Q: What is an EPG?

A: An EPG is a program guide describing which programs are being aired on a TV channel for a given period.

Input Data

The input JSON represents the EPG for a given period. It consists of keys indicating the day of a week and the corresponding programs start and end time for that day. The JSON includes data for one TV channel:

```
{
      <dayofweek>: <program air time>
      <dayofweek>: <program air time>
      ...
}
```

- **<dayofweek>**: monday / tuesday / wednesday / thursday / friday / saturday / sunday
- **<program air time>**: an array objects containing air time and the program title. Each object consists of 3 keys:
  - **title**: the title of the program
  - **state**: begin or end of the program
  - **time**: beginning/ending time as unixtime (1.1.1970 as the date)

Example of a program (Nyhederne) being aired on Monday from 13 to 13:30:

```
{
  "monday": [
    {
      "title": "Nyhederne",
      "state": "begin",
      "time": 46800
    },
    {
      "title": "Nyhederne",
      "state": "end",
      "time": 48600
    }
  ]
}
```

Special Cases

- If no programs are being aired for the entire day, the array of air times is empty
    - “monday”: [] means no programs are being aired on monday
- A program might begin one day and end the next day, e.g. a program might begin on thursday and end on friday
    - In that case the “thursday” object contains the beginning state and “friday” object contains the ending state for that program
    - When displaying air times for programs which span over two days, ending time is displayed as part of the day the programs beigns.
        - E.g. Thursday: ESLPro 23 – 1
- A program can be aired (beginning and ending) multiple times during the same day
    - E.g. on wednesday Nyhederne is aired from 6 to 12 and again from 21 to 21:30

Example:

```
{
  "wednesday": [
    {
      "title": "Nyhederne",
      "state": "begin",
      "time": 21600
    },
    {
      "title": "Nyhederne",
      "state": "end",
      "time": 43200
    },
    {
```

Output for this input would be:

```
Wednesday: Nyhederne 6 – 12, 21 – 21:30
Thursday: ESLPro 23 – 1
Friday: Nothing aired today
```

```
      "title": "Nyhederne",
      "state": "begin",
      "time": 75600
    },
    {
      "title": "Nyhederne",
      "state": "end",
      "time": 77400
    }
  ],
  "thursday": [
    {
      "title": "ESLPro",
      "state": "begin",
      "time": 82800
    }
  ],
  "friday": [
    {
      "title": "ESLPro",
      "state": "end",
      "time": 3600
    }
  ]
}
```

Deliverables

Part 1

Build a HTTP API that accepts EPG data as input in JSON and returns a human readable version of the data using 24-hour clock.

Output example could be:

Monday: Nyhederne 6 - 8:30 / Dybvaaaaad 10 - 10:30
Tuesday: Nothing aired today
Wednesday: Fodbold 14:30 - 16
Thursday: ESL 22 - 2
Friday: Nothing aired today
Saturday: Comedy 10 - 12:30, 15 - 17:30 / Fodbold 18 - 21:30
Sunday: Nybyggerne 10:30 - 13

Your API can print the formatted output to the console or return it as the response to the caller of the API.

Part 2

Tell us what you think about the input data model. Is the current JSON format a good fit for this kind of data or can you come up with a better version? There are no right answers here. Please write your thoughts in the *readme.md*

Full JSON Example

```
{
  "monday": [
    {
      "title": "Nyhederne",
      "state": "begin",
      "time": 21600
    },
    {
      "title": "Nyhederne",
      "state": "end",
      "time": 36000
    },
    {
      "title": "Dybvaaaaad",
      "state": "begin",
      "time": 36000
    },
    {
      "title": "Dybvaaaaad",
      "state": "begin",
      "time": 38100
    }
  ],
  "tuesday": [],
  "wednesday": [
    {
      "title": "Nyhederne",
      "state": "begin",
      "time": 21600
    },
    {
      "title": "Nyhederne",
      "state": "end",
      "time": 43200
    },
    {
      "title": "Fodbold",
      "state": "begin",
      "time": 50400
    },
    {
      "title": "Fodbold",
      "state": "end",
      "time": 55800
    },
    {
      "title": "Nyhederne",
      "state": "begin",
      "time": 75600
    },
    {
      "title": "Nyhederne",
      "state": "end",
      "time": 77400
    }
  ],
  "thursday": [
    {
      "title": "ESL",
      "state": "begin",
      "time": 43200
    },
    {
      "title": "ESL",
      "state": "end",
      "time": 46800
    },
    {
      "title": "ESLPro",
      "state": "begin",
      "time": 82800
    }
  ],
  "friday": [
    {
      "title": "ESLPro",
      "state": "end",
      "time": 3600
    }
  ],
  "saturday": [
    {
      "title": "Comedy",
      "state": "begin",
      "time": 52200
    },
    {
      "title": "Comedy",
      "state": "end",
      "time": 59400
    },
    {
      "title": "Nybyggerne",
      "state": "begin",
      "time": 81600
    }
  ],
  "sunday": [
    {
      "title": "Nybyggerne",
      "state": "end",
      "time": 5400
    },
    {
      "title": "Dybvvvvvad",
      "state": "begin",
      "time": 41400
    },
    {
      "title": "Dybvvvvvad",
      "state": "end",
      "time": 43200
    }
  ]
}
```

Output

```
Monday: Nyhederne 6 – 10 / Dybvaaaaad 10 – 10:35
Tuesday: Nothing aired today
Wednesday: Nyhederne 6 – 12, 21 – 21:30 / Fodbold 14 – 15:30
Thursday: ESL 12 – 13 / ESLPro 23 - 1
Friday: Nothing aired today
Saturday: Comedy 14:30 – 16:30 / Nybyggerne 22:40 – 1:30
Sunday: Dybvaaaaad 11:30 - 12
```