Frontend Engineering Challenge

Here at Eight, we ingest terabytes of sleep data every week. Every Eight device sends data constantly, that data is absorbed in the data collection pipeline and then our algorithms classify for each user whether that person was in bed, sleeping, which stage of sleep etc. The algorithms are smart enough to group together sleep times of a person into one “session” (also known as interval), if that makes sense (let’s say someone woke up for a few minutes in the middle of the night).
Our users view their sleep sessions data via a mobile app.

The Challenge

Your task is to build a web app to visualize sleep data collected by Eight.

You have access to sleep session data in JSON format (see below) for three different users: the web app should display the sleep data of a user in a meaningful way, with the ability to switch the currently displayed user.

Data Format

You are provided with 3 files (hosted in S3) that contain the data to visualize.

1. Each input file holds an object with an “intervals” property which contains an array of sleep session information (sleep intervals).
2. Each interval has a unique ”id” and the time when the interval started “ts”
3. Each interval has an array of stages, each stage has a duration (in seconds) and the type of stage, which can be one of the following:
    a. “out” - user is out of bed
    b. “awake” - user is in bed, but not asleep
    c. “light” - user is in light sleep
    d. “deep” - user is in deep sleep
4. Each interval has property named “timeseries” which is a map from a string (name of timeseries) to an array of timeseries data. The keys of the timeseries object are:
    a. “tnt” - toss and turns
    b. “tempRoomC” - room temperature, in celsius
    c. “tempBedC” - bed temperature, in celsius
    d. “respiratoryRate” - in breaths per minute
    e. “heartRate” - in beats per minute

You can find an annotated example of the data format here:
https://gist.github.com/maghis/8c35fe1bb5c7810bdcc6ca389c6cd702

Following are links to each data file:
https://s3.amazonaws.com/eight-public/challenge/2228b530e055401f81ba37b51ff6f81d.json
https://s3.amazonaws.com/eight-public/challenge/d6c1355e38194139b8d0c870baf86365.json
https://s3.amazonaws.com/eight-public/challenge/f9bf229fd19e4c799e8c19a962d73449.json

General Guidelines

1. Feel free to use any language, infrastructure or technology, as long as we can run the full stack on our machines.
2. Your deliverable will be a public github repository with the source code and a brief readme with instructions on how to run it.

Bonus points

1. Do not retrieve the json files directly in the frontend, but use the backend to get them.
2. All the rendering happens in the client.
3. Use of a config file (Not a lot to configure? Think of at least one thing you can read from a file).
4. Create a Dockerfile at the root of your code that we can run in order to create a docker that runs your program.