awake labs

Awake Labs - Software Development Intern Coding Challenge

Background:
The Awake Labs wearable app built for the Samsung Galaxy watch collects data at an approximate rate of 3 data points per second. This data is stored on the watches' internal database, and a JSON package is transmitted to our PostgreSQL backend database hosted on AWS every minute via WiFi. The payload of this package includes the following:

```
var payload = {
        watchIdentifier:watchIdentifier,
        anxietyState: anxietyState,
        watchTime: Date.now(),
        type: 'ONLINE',
        data: {
            currentBpm: currentBpm,
            anxietyLevel: getAnxietyLevel(),
            baselineProgress: getBaselineProgress(),
            heartRates: heartRatePackage,
            time: Date.now(),
            state: {
                batteryLevel: batteryLevel
            }
        },
    };
```

Description of payload:

Name	Description	Relevant to challenge?
watchIdentifier	Unique identifier used to link each watch to its relevant user	NO
anxietyState	Normal/Low/Medium/High. Calculated using a simple logic with 'anxietyLevel' as the input	NO
watchTime	Time recorded on the watch at the moment the payload was sent	NO
type	Was the watch connected to the internet or not at the time the payload was sent?	NO
data	See description below in the Coding Challenge Setup section	YES

Coding Challenge Setup

We have created a JSON file with realistic synthetic data that is structurally identical to our production data. This corresponds to the ‘data’ part of the payload that is sent from the watch to the PostgreSQL backend database hosted on AWS. For example:

```
"currentBpm":114,
"anxietyLevel”:0,
"baselineProgress":100,
"heartRates":
        [{"time":1554326492237,
        "rrInterval":0.758,
        "heartRate":104,
        "anxietyLevel":0,
        "motion":false},

        {"time":1554326495239,
        "rrInterval":1.154,
        "heartRate":104,
        "anxietyLevel":0,
        "motion":false}],

"time":1554326495239,

"state":{"batteryLevel":0.96}
```

Name	Description
currentBPM	Beats per minute (BPM). Samsung has a built-in signal processing algorithm in its OS that converts the raw PPG signal into BPM. The BPM value here is the value captured at the time the package was sent.
anxietyLevel	This is calculated by the Holland Bloorview (HB) algorithm. This is the anxietyLevel value at the time the package was sent
baselineProgress	The HB algorithm uses a baseline to ascertain the anxietyLevel. This baseline starts when the watch is turned on
heartRates	This nested JSON includes the time at which the data point is captured, rrInterval (this is a feature of the heart rate signal), heartRate (i.e. BPM), anxietyLevel (calculated by HB algorithm) and motion (true or false, from Samsung OS).
time	The is the time at which the package was sent (in unix time)
state	The battery level at the time that the package was sent

Your challenge:

Create a time-series graph with the synthetic data we provided. Show time in human-readable format on the x-axis in 1 hour increments. You can choose which data to plot on the y-axis of your graph. For example, you can display just the anxietyLevel over time, or you can choose to also include currentBPM.

The synthetic data is formatted as a nested JSON. We will be looking closely at how you decide to handle this data structure when creating your graph. You should think about how to structure the data to optimize for speed & efficiency.

Requirements

1. Create your graph using either a Jupyter notebook (or similar tool) or a web app that works when run on localhost.
   - Make sure to include instructions to help us run your Jupyter notebook and/or web app.
2. Your graph should allow us to manipulate the time range and view the data over a variety of ranges, like 24 hours, 1 week…
3. Comment your code and include a write up to explain your approach to handling the data in the nested JSON.

Deliverable:

Put your code up on GitHub or Bitbucket along with any documentation you create. Include instructions to run your code and your write up. We will be paying close attention to your approach to handling the nested JSON and the performance of your graph.

Want to really impress us?

If you want to really impress us, create a second visualization and/or add some additional features to your time-series graph.

After submitting your coding challenge, we will schedule a 1-hour discussion where you can talk us through your thought process and design decisions and answer questions from our team.

Evaluation Rubric - Scored on a scale of 1-5, where 1 = Does not meet criteria and 5 = exceeds expectations

Item	Description
Creativity	● Referring to your approach to handling the data in the nested JSON ● What technologies did you choose to create your graph (libraries, 3rd party services, etc.)?
Code Quality	● What is the overall quality of your code? ● How quickly does your graph respond when manipulating time ranges?
Communication	● How well is your code commented? What is the quality of your write up? ● During the 1-hour discussion, how well do you explain your decisions & answer questions?