

Backend Interview: Take-Home Exercise

Brightwheel Take-Home Exercise

Thanks for your interest in Brightwheel!

We ask that you limit yourself to a maximum of two hours when working on this.

If you run out of time, don't worry - add some notes in your README to describe what you would have worked on given more time.

Summary

For this exercise, imagine that we have devices that record an integer at arbitrary intervals called "readings." These readings are then sent out over the internet as HTTP requests. We would like you to implement a simple web API that receives these requests.

Requirements

Recording Readings

Your API service should accept `POST` HTTP requests from these devices. You are free to define this `POST` route as you see fit. Request bodies will contain JSON payloads in the following format:

```
{
  id: (string) The device identifier. This will be a uuid.
  readings: [
    {
      timestamp: (string) Timestamp for the reading
      in [iso 8601 format](https://en.wikipedia.org/wiki/ISO_8601)
      count: (integer)
    }
  ]
}
```

Example Request:

```
{
  "id": "36d5658a-6908-479e-887e-a949ec199272",
  "readings": [
    {
      "timestamp": "2021-09-29T16:08:15+01:00",
      "count": 2
    },
    {
      "timestamp": "2021-09-29T16:09:15+01:00",
      "count": 15
    }
  ]
}
```

Note that a single request can account for multiple readings.

Returning Data

It should also expose HTTP endpoints that return the following information:

- The most recent reading for a given device id, sorted by timestamp.
- The cumulative count for a given device id.

You are free to define the HTTP routes and response payloads for these as you see fit.

Other Requirements

Programming Language

Use any programming language you like.

Device Requests

These devices are unreliable. As such:

- Readings may be sent out of order.
- Devices will occasionally send `POST` requests with duplicate payloads. Duplicate payloads should be ignored.
- Both `POST` request bodies and individual readings may be malformed. Malformed readings should be ignored.

Data Persistence

Please do not use a database. An in-memory only solution is fine. If you persist data to disc, use a plain-text file (json, CSV, etc.)

Third-Party Libraries

Other than the database requirement above, feel free to use any third-party libraries you deem appropriate. If you do, make sure to include instructions on how to install them in your README.

README

Please include a `README.md` file with your exercise. It should include:

- Instructions on how to start up your application
- List the HTTP endpoints exposed by your application and any required parameters
- Any notes on design decisions
- Anything you would have included given more time