Fetch Rewards Coding Exercise - Backend Software Engineering

What do I need to submit?

Please write a web service that accepts HTTP requests and returns responses based on the conditions outlined in the next

section. You can use any language and frameworks you choose.

We must be able to run your web service; provide any documentation necessary to accomplish this as part of the repository you submit.

Please assume the reviewer has not executed an application in your language/framework before when developing your documentation.

What does it need to do?

Background

Our users have points in their accounts. Users only see a single balance in their accounts. But for reporting purposes we actually track their points per payer/partner. In our system, each transaction record contains: **payer** (string), **points** (integer), **timestamp** (date).

For earning points it is easy to assign a payer, we know which actions earned the points. And thus which partner should be paying for the points.

When a user spends points, they don't know or care which payer the points come from. But, our accounting team does care how the points are spent. There are two rules for determining what points to "spend" first:

- We want the oldest points to be spent first (oldest based on transaction timestamp, not the order they're received)
- We want no payer's points to go negative.

We expect your web service to

Provide routes that:

- Add transactions for a specific payer and date.
- Spend points using the rules above and return a list of `{ "payer": <string>, "points": <integer> }` for each call.
- Return all payer point balances.

Note:

- We are not defining specific requests/responses. Defining these is part of the exercise.
- We don't expect you to use any durable data store. Storing transactions in memory is acceptable for the exercise.

Example

Suppose you call your add transaction route with the following sequence of calls:

- `{ "payer": "DANNON", "points": 1000, "timestamp": "2020-11-02T14:00:00Z" }`
- `{ "payer": "UNILEVER", "points": 200, "timestamp": "2020-10-31T11:00:00Z" }`
- `{ "payer": "DANNON", "points": -200, "timestamp": "2020-10-31T15:00:00Z" }`
- `{ "payer": "MILLER COORS", "points": 10000, "timestamp": "2020-11-01T14:00:00Z" }`
- `{ "payer": "DANNON", "points": 300, "timestamp": "2020-10-31T10:00:00Z" }`

Then you call your spend points route with the following request:

```
{ "points": 5000 }
```

The expected response from the spend call would be:

```
[
    { "payer": "DANNON", "points": -100 },
    { "payer": "UNILEVER", "points": -200 },
    { "payer": "MILLER COORS", "points": -4,700 }
]
```

A subsequent call to the points balance route, after the spend, should returns the following results:

```
{
    "DANNON": 1000,
    "UNILEVER": 0,
    "MILLER COORS": 5300
}
```

How do I submit it?

Provide a link to a public repository, such as GitHub or BitBucket, that contains your code to your recruiter.

FAQ

How will this exercise be evaluated?

An engineer will review the code you submit. At a minimum they must be able to run the service and the service must provide the expected results. You should provide any necessary documentation within the repository. While your solution does not need to be fully production ready, you are being evaluated so put your best foot forward.

I have questions about the problem statement

For any requirements not specified via an example, use your best judgement to determine the expected result.

Can I provide a private repository?

If at all possible, we prefer a public repository because we do not know which engineer will be evaluating your submission. Providing a public repository ensures a speedy review of your submission. If you are still uncomfortable providing a public repository, you can work with your recruiter to provide access to the reviewing engineer.

How long do I have to complete the exercise?

There is no time limit for the exercise. Out of respect for your time, we designed this exercise with the intent that it should take you a few hours. But, please take as much time as you need to complete the work.