Recruitment case: Backend Developer - Using DI Spider API

Background

The aim of this task is to assess the way you approach problems, including your coding style, expertise and willingness to experiment, as well as provide us with a common ground for a technical interview.

We'd love to see what kind of solution you come up with and how you approach problem solving.
There is no hard time limit set for this task, but generally one evening should be the goal. Due to time constraints, we don't expect all the bells and whistles and you're encouraged to focus on your core strengths and things that you think are important for production service — just leave notes and TODOs if there are parts of implementation you didn't manage to complete. You will be using DI's API in a limited time, namely our route Optimization API called Spider. Your feedback is important to us, so let us know what you think about the task — before you started or after you're done ☺

Preparations

Our customers are really dependant of our systems to be able to do their deliveries. In order for them to do this in a best possible way we need to send their customers through a route optimization model which is a solution called Spider. In this case you will be given a CSV file containing all the latitude and longitude of the customers locations. These needs to be sent over to the optimization engine and obtained back, stored and returned to the customer in a good manner. Your choice of storage and how you process the file is optional (if you read file or send through an API i.e).

Your application is required to have the following features:

- Receive data from the file
- Send the coordinates to our optimization engine
- Evaluate the response limits and get the solution from optimization engine
- Store and present the data to a customer (no frontend needed to be built so API call is sufficient)

Using Java, implement a RESTful API service to receive a data file with latitude, longitude. The service should connect to some sort of storage device to store the input data and to store the optimized data.

Some general things to consider:

- Data modeling.
- API design. How should it be presented in the API? Will it be easy to include more queries or extend the current ones to cater for new feature requests?
- Would it be easier for other developers to get up-and-running if you use docker?
- Testing approach. How would you approach it in your implementation?

Final result should consist of:

1. Source code with instructions on how to run it in a git repository we can access (Github, Bitbucket etc.)
2. A service deployed to a cloud provider of your choice using IaC approach.
   a. This is optional — only do it if you would like to demonstrate your DevOps skills.

Expected time to implement

There is no expectation to use a lot of time. Typically one evening should be sufficient to complete the task at a sufficient level. We do not expect all the bells and whistles to be present, but during the second interview feel free to elaborate on any ideas you might have had during the task that you were not able to implement.

How we evaluate

We have a second interview where we have looked into your case on Github. During the second interview we would like you to present your case and talk us through how you were thinking during the task. We would like to know why you chose to do the things you chose. We would also like to hear if you were faced with any difficulties that required you to do some extra thinking. We will then based on your task and presentation evaluate it as a whole.

Technical details

Documentation of the Spider optimization API:

Welcome to the Vehicle Routing Service

When you create a session you will need to send in some additional data besides deliveries, use the following JSON from input.json

To be able to use Spider you will need to create a session, wait until it is ready. When optimization is running, you can check the bestSolutionValue, and choose to stop at your request.

In the create sessions endpoint the `orders[].delivery.address` field should consist of a coordinates on the following format: `“lon=12,3;lat=4,56"` for one single address. The ID can be a unique ID you provide yourself. Below is an example order:

```
"orders": [
      {
        "id": "3316602",
        "type": "delivery",
        "size": [
          1
        ],
        "delivery": {
          "address": "lat=56.9148408831298;lon=13.9884584483723"
        }
      },
```

Fullstack: UI for spider API

You are to create a UI version where the customers can upload their file to your server and to be able to see the result of the optimization. If the main system is down, the customers expect to be self dependant. They expect to be able to upload their file of customer delivery points, and get a response that they are able to provide to their drivers. In this case you are tasked to create a simple frontend UI to the backend created earlier where the customers can log in, upload their file and get a result back that the drivers could understand. and drive by.

Preparation

Your UI is required to have the following features:

- Simple login - Provide a way of logging in
- Upload the file
- Receive the response from the backend system presented in a way that drivers can understand

Tools to use:

- React
- Typescript

Good luck!