

Scheduler Client Java Task:

You must use: java (1.8) and maven 3

You may use: any additional dependencies as you like. (Hint: you would do well to use at least a few)

1) The Zoomcare scheduler, available when you go to zoomcare.com is powered by a rest endpoint that returns a list of clinic schedule data. Create a client that consumes this endpoint and unmarshals it into a java objects that each represent a clinic with these properties:

- A) Clinic hours
- B) Name
- C) Clinic code

The endpoint is a POST HTTP Request to:

`https://www.zoomcare.com/zoom-api/schedule`

Sample POST body:

```
{
  "serviceLineIds": ["e675e4e7-978a-4c89-ad09-63b30965821f"],
  "latLong": "(45.5235,-122.676)",
  "duration": 15,
  "clinicId": "",
  "providerId": "",
  "orderedLabId": "",
  "page": 1,
  "scheduleDate": 0,
  "allClinics": false
}
```

You'll also need to send the scheduleDate which is the current date/time in milliseconds since the epoch. You can find some examples of inputs and outputs on the zoomcare.com website.

2) Use the client you created to make a command line program that displays a numbered ordering of clinic codes and when a number is entered the open and close time of the current day are displayed. The "dayOfWeek": 1 represents Sunday. Include instructions for running your program.

```
1: pd
2: nw
3: gr
...
```

> 1

>open at 8am; closed at 10pm

System security is very important to us and certain file extensions will be blocked for security purposes, resulting in delays to your application. We need to be able to run and build your code ourselves, so please submit your code as a zipped file of source code and supporting files.

Please include a brief explanation of your design and assumptions, along with your code, as well as detailed instructions to run your application.

We assess a number of things including the design aspect of your solution and your object oriented programming skills. While these are small problems, we expect you to submit what you believe is production-quality code; code that you'd be able to run, maintain, and evolve. You don't need to gold plate your solution, however we are looking for something more than a bare-bones algorithm.

We want our hiring process to be fair, and for everyone to start from the same place. To enable this, we request that you do not share or publish these problems.