Part One

Skills:

1. Spring Boot framework
2. Java

Designs:

1. Through requesting the API (https://interview-interval-service.dev-svcs.enernoc.net/v1/meter), we firstly get the list of meter_id, earliest_time and latest_time. Secondly, requesting the Interval API to get all the data in one day from certain meter. Thirdly, computing the maximum value from the data per meter per day, and comparing the data with the data of other days with the same meter. Fourthly, computing the most maximum value data of one meter, and then store the data in the map with meter_id as key and day and maximum_value as value.
2. In order to promote the computing performance, thread pooling technology is used to realize multithreading.
3. A Restful API is created to let the other services and the front-end button to activate and trigger.

Part Two

Skills:

1. Docker

Designs:

1. Through Dockerfile, the Part One program can run in a docker.
2. If the two instances in the requirement mean two services, two dockers can be used to run the Part One program. However, in that case, the meter API will be requested twice, the solution to me is the RDS database.
3. Two table are established as Table Meter (meterId as primary key, earliest, latest and mark) and Table Interval (meterId as primary key, intervalDttm and intervalMaxValue). The mark field stands for the status of the meter data, and 0 standing for unprocessed, 1 standing for processing, 2 standing for processed and 3 standing for processing failed. When these two instances are computing, they each will select one piece of data. When they are calculating, the mark filed will update to processing, and after calculating it will update to processed. In order to enhance robustness, if timeouts the mark field will update to processed failed, the program will retry the processed failed data. This is my idea to this solution, and I haven't finished.

Part Three:

Skills:

1. Angular

Designs:

1. In the Angular demo, I change some parameters and get two value from exposed API (/v1/metric). Two "AtomicLong" data type global variables are used to ensure thread safe.
2. The Angular program contains trigger and getMetric method, and call the getMetric method per second to update the page.

Operating manuals

Backend

```
1.  ./mvnw clean package
    // put interval-service-1.0.0.jar and Dockerfile in the same directory, and executive the following command in the current directory
2.  docker build --build-arg JAR_FILE=interval-service-1.0.0.jar -t interval-service:1.0.0
3.  docker run -p 8080:8080 interval-service:1.0.0
```

Frontend

```
1.  npm install
2.  ng serve --proxy-config proxy.conf.json
```