Tech - Onsite Interview - Data Engineer/Scientist (Junior)

Purpose:

- To allow us to know how fast you can learn new concepts, technologies and how resourceful you are

Estimate time: 2 hrs

Objectives:

You are given a file within the same folder as this question:

- raw-bid-win.tar.gz

WRITE A PROGRAM in Java/Scala/Python2/NodeJS:

1) Extract and parse the files in the raw-bid-win folder(which contains lines JSON string) into JSON objects(with the schema described below), then insert them into MongoDB under collection `individualWins` of database `analyticsInterview`.

```
auctionId (Self explanatory and found easily in the JSON object, default
value: "NA")
 campaignId (Found under `biddingMainAccount` of the JSON object, default
value: "NA")
 creativeId (Found under `bidResponseCreativeName` of the JSON object,
default value: "NA")
 adgroupId (Found under `biddingSubAccount` of the JSON object, default
value: "NA")
 userAgent (Found under `bidRequestString->userAgent` of the JSON object,
default value: "Others")
 site (Found under `bidRequestString->url` of the JSON object, default
value: "Others")
 geo (Found under `bidRequestString->device->geo->country` of the JSON
object, or under `bidRequestString->device->ext->geo_criteria_id` if the
former cannot be found, default value: "Others")
 exchange (Found under `bidRequestString->exchange` of the JSON object,
default value: "Others")
 price (Found under `winPrice` of the JSON object, strip out USD/1M and
store as numerical format, default value: 0)
 time (Found under `bidRequestString->timestamp` of the JSON object.
Convert the value into Unix Timestamp at the start of the hour, default
value: 0)
```

- If you are unable to parse the line, you may ignore the line
- Default value refers to the value you can use if you are unable to parse the key for whatever reason

2) Group and aggregate the data stored in `individualWins` with the following requirements and upsert into 'geoAggregation' based on Group Key

Group Key	Values to Aggregate	Collection to store the results into	Schema

• campaignId • creativeId • adgroupId • geo • time	• Sum of `price` • Minimum of `price` • Maximum of `price` • Total entries per group key	geoAggregation	• campaignId • creativeId • adgroupId • geo • time • totalPrice • minPrice • maxPrice • totalCount

3) Dump out the data in JSON format based on the results from #2 and submit both the code and data dump to us.

(see overleaf for sample dump)

Sample of an article document

```
{
  "_id": ObjectId("5800808130901b477ba6f1ef"),
  "adgroupId": "strategy",
  "campaignId": "97e53d9f364980c19b1d018a9ac07eb97d04d804",
  "creativeId": "knxad_knx4605_201603092894",
  "geo": "FJI",
  "time": 1470279600,
  "totalCount": 13,
  "totalPrice": 1000000,
  "minPrice": 20,
  "maxPrice": 100000
},
{
 ...
}
```