# **Software Engineering challenge**

Welcome to this challenge. This exercise is a small example of some of the problems you could actually encounter working with us. As with most real problems, the time spent on them can range from a few hours for a proof of concept, to a lot more when developing a production-ready system.

We expect this exercise to be completed in around 3 or 4 hours. It is usually not enough time to get a flawless solution, and that is taken into account while we are reviewing it. Even if not a perfect solution is implemented, it would be great if you can write a small summary of the assumptions you took, and also what the future work would be for problems you found, or to optimize your solution.

Good luck and have fun with the challenge!

## **Business Model**

Advertisement banners are displayed to users in a mobile application (app\_id) in a country (country code) from an advertiser (advertiser\_id). When this happens, an impression event is recorded and stored. Optionally, if the user clicks on the banner, a click event is recorded. Revenue is generated only in the case of a click being triggered.

## **Input**

### **Arguments**

Your application should accept 2 lists of file names with click and impression events.

### **Impression event schema**

* id (string): a UUID that identifies the impression.
* app\_id (integer): an identifier of the application showing the impression.
* country\_code (string): a 2-letter code for the country. It doesn't necessarily comply with any standard like ISO 3166.
* advertiser\_id (integer): an identifier of the advertiser that bought the impression.

Example data can be found on impressions.json.

### **Click event schema**

* impression\_id (string): a reference to the UUID of the impression where the click was produced.
* revenue (double): the quantity of money paid by the advertiser when the click is tracked.

Example data can be found on clicks.json.

## **Goals**

### **1. Read events stored in JSON files**

Read and parse the events for both impressions and clicks from the provided JSON files in your entry point. Some events may not comply with the provided schema. You can use the library of your choice to perform the JSON parsing.

### **2. Calculate metrics for some dimensions**

The business team wants to check how some metrics perform depending on a few dimensions. For example, they would like to check how applications are performing depending on the country. This will be very useful for them, as they will be able to spot new opportunities or countries that are performing poorly.

Metrics:

* Count of impressions
* Count of clicks
* Sum of revenue

Dimensions:

* app\_id
* country\_code

Please, write the output to a JSON file using the following format:

[

{

"app\_id": 1,

"country\_code": "US",

"impressions": 102,

"clicks": 12,

"revenue": 10.2

},

...

]

### **3. Make a recommendation for the top 5 advertiser\_ids to display for each app and country combination.**

Now, the business team wants to know which are the top advertisers for each application and country. This will allow them to focus their effort on these advertisers. To measure performance, we will check for the highest rate of revenue/impressions. That is, the advertisers that, on average, pay more per impression.

Output fields:

* app\_id
* country\_code
* recomended\_advertiser\_ids (list of top 5 advertiser ids with the highest revenue per impression rate in this application and country).

Please, write the output to a JSON file using the following format:

[

{

"app\_id": 1,

"country\_code": "US",

"recommended\_advertiser\_ids": [32, 12, 45, 4, 1]

}

]

## **Technical requirements**

* Write your application using the Java programming language. You can choose the build tool of your choice.
* You can use a library of your choice to parse JSON and program arguments.
* You can use a DBMS like mysql to save data and use SQL to generate reports.
* Please, don't use any data processing framework (Spark, Flink, Akka...) for goals 1 and 2. You can use them for the third one.
* Your application will be running on a single instance with 8 cores.

Please, provide your code as a git repository with a README on how to execute it with the sample files.