Please create github repository and publish the code there.

Task (Should take 3-6 hours)

The goal of this task is to analyze and transform the > 500.000 reviews from amazon.

Please go to Kaggle.com and download (please take the 300MB csv file):

<https://www.kaggle.com/snap/amazon-fine-food-reviews>

Please note how much time did you spend doing that task. **If you don’t have time to implement something just comment it** so we know that you are aware of limitation. If you have questions please ask us but you can also take initiative it is enough that you comment what you decided.

Please use any JVM language. You are free to choose any frameworks and libraries that make you productive in solving the problems below.

We are interested in:

1) Finding 1000 most active users (profile names)

2) Finding 1000 most commented food items (item ids).

3) Finding 1000 most used words in the reviews

4) We want to translate all the reviews using Google Translate API. You can send up to 1000 characters per HTTP call. API has 200ms average response time. How to do it efficiently and cost effective (you pay for API calls and we have concurrency limits - 100 requests in parallel max) - please mock the calls to google translate API.

We assume that the endpoint will be: <https://api.google.com/translate>

The content type: application/json and the example body of the POST:

```

{input\_lang: ‘en’, output\_lang: ‘fr’, text: “Hello John, how are you?”}

```

Output:

```

{text: ‘Salut Jean, comment vas tu?’}

```

Any errors will be reported by the HTTP codes.

* We are interested in clean testable code (add tests if you have time).
* How do you make sure that there are no duplicates in the file?
* We are interested in using full multi core CPU power.
* We will be running this on machine with 500MB of RAM. How do you make sure that we are not using more than that? How are you going to monitor the memory usage of your program?
* Our goal is to support the files with up to 100M reviews on multiple machines with 500MB of RAM and 4 core CPUs. How are you going to make it happen?

Please provide working code (command to compile and run the program) that prints output of point: 1,2,3 to standard output sorted alphabetically and executes point 4 to mocked endpoint (when launched with the argument ‘translate=true’).