

# Expertise test - BE Java Engineer (v3.0)

## Data set:

You will receive some CSV files along with this document (e.g. "LD\_A2302900\_20190301\_20190401.csv"). These documents contain vehicle telemetry data - like vehicle speed, position, RPM, etc. We get these files from an external data source.

Each file contains a telemetry log for a single vehicle for one day.

You will notice that these files have different data sets. This is because some files are for *tractors* and some for *combines*. Files that start with "LD\_A" are for tractors and "LD\_C" for combines.

## Instructions:

We would like you to create a RESTful service, which will enable users to import such files into the DB and then query (filter) the data.

The user would like to filter by any field in the data set (either from tractor or from combine).

For example, this is how the request parameters could look like:

```
[
  {
    "field": "GrainTankUnloading",
    "value": true
  },
  {
    "field": "GroundSpeed",
    "operation": "GreaterThan",
    "value": 4.50
  }
]
```

Which would mean: Return all the data (from tractors and combines) where GrainTankUnloading is true and GroundSpeed is greater than 4.5.

We would like you to support the following operations: "Equals" (default), "LessThan", "GreaterThan" and "Contains". The implementation should validate the input and reject inapplicable combinations, e.g. not all operations apply to every field type (e.g. "Contains" on a field of type Integer).

## Prerequisites:

You should use Java or Kotlin and whatever related technologies/frameworks.

## Delivery:

We would like you to host your code in a git repository (e.g. github) and provide a readme file with the following:

- short description of the solution,
- how to run and use it.

Imagine you are handing your solution over to the end customer, hence you want to provide a code you'll be proud of.

Of course we don't want you to over-engineer this and spend too much time on it, so If you see some steps where you could further improve your code, write it in a comment or a readme. In the end we want to see how you structure and write the code, so try to include as much as you can. Other things we can discuss over a live tech interview.