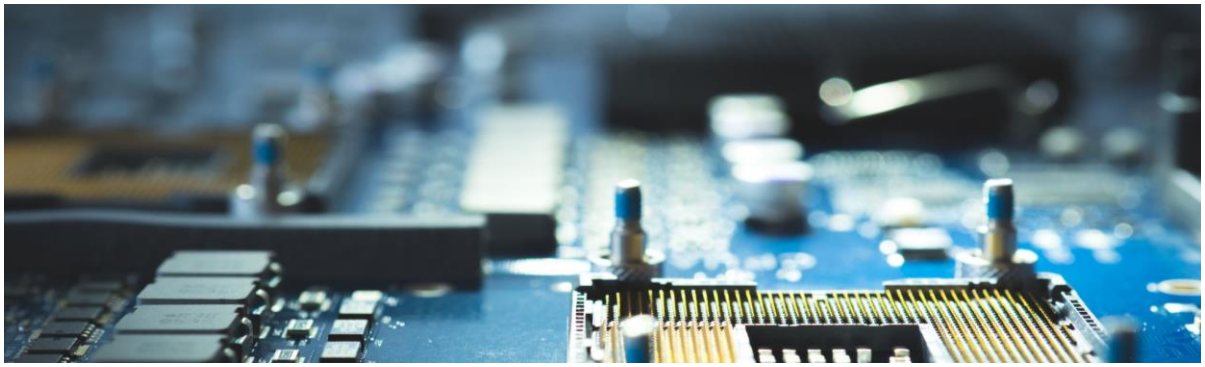

Hertz Data Engineer Assignment

Thank you for your interest in Hertz. The second part of our process entails a small test and gives us a good baseline of your technical ability. We'd like to invite you to participate, and the process is outlined below. It would be great to have your work back within one week upon receiving this mail, however, if you need more time, please let us know.

Project brief

One of our backend applications sends various events related to location and diagnostics of numerous cars available in our fleet systems.

The events are classified as below:

LOCATION – contains geolocation coordinates, i.e. Latitude, Longitude.
DIAGNOSTICS – contains various kinds of readings from different sensors in the vehicle. These sensors are as follows:

- MILEAGE – contains odometer values (in meters).
- FUEL – contains fuel values (in % of total volume).
- TIRE – tire pressure corresponding to various tires.
- BATTERY – vehicle battery voltage.

These events are periodically uploaded to our datalake in JSON format. The volume of the events received are in the range of several hundreds of gigabytes/day. The data science and analytics team want to be able to query these events, so they can analyse trip behaviour (based on location data) and car usage (based on diagnostics data) and answer various business questions, e.g. "How many vehicles have oil levels below 10% on 2nd January 2022 between 3 pm – 4 pm?"

For this challenge, you (the data engineer) have been requested to perform the below:

1. Create a PySpark script that can run in a Spark Cluster
2. The script must read and process the data in JSON format using PySpark
3. Write the processed files in an efficient format to local machine

You are free to choose the output format and organising of the processed data but please explain your decision.

Dataset
The attached file – **events.zip** contains several event log files. The files are in JSON lines format and look something like this:

Location Event Example


```
{
  "timestamp":"2022-01-02T15:04:22.000000Z",
  "type":"LOCATION",
  "eventId":"ae617df8-f69273",
  "eventData":{
    "latitude":"74.21441",
    "longitude":"80.7320"
  },
  "vehicleId":"629339969"
}
```


Diagnostic Event Example


```
{
  "timestamp":"2022-01-02T15:04:30.657000Z",
  "type":"DIAGNOSTIC",
  "eventId":"05bfbba2-450648",
  "eventData":{
    "sensor":"OIL",
    "sensorData":{
      "remainingLife":17.0
    }
  },
  "vehicleId":"478553739"
}
```


You are free to use helper libraries; we don't expect you to write everything from scratch.

The project should be a zipped git repository excluding input and output files. Commits should be made at logical intervals, with appropriate comments on work. We value clean, testable code accompanied by documentation on how to set up and verify the solution.

Feel free to respond with any questions; we're excited to see your work!