Coding Assignment 2

Requirements

- Java 8
- Use of any open-source library is allowed
- Program must use gradle build system to resolve dependencies, build and test

Summary

Our custom-build server logs different events to a file. Every event has **2** entries in log - one entry when the event was started and another when the event was finished. The entries in log file have no specific order (it can happen that finish event is logged before event start)

Every line in the file is a JSON object containing event data:

- **id** - the unique event identifier
- **state** - whether the event was started or finished (can have values "STARTED" or "FINISHED")
- **timestamp** - the timestamp of the event in milliseconds

Application Server logs also have the additional attributes:

- **type** - type of log
- **host** - hostname

Example:

```
{"id":"scsmbstgra", "state":"STARTED", type:"APPLICATION_LOG",
host:"12345", "timestamp":1491377495212}
{"id":"scsmbstgrb", "state":"STARTED", "timestamp":1491377495213}
{"id":"scsmbstgrc", "state":"FINISHED", "timestamp":1491377495218}
{"id":"scsmbstgra", "state":"FINISHED", type:"APPLICATION_LOG",
host:"12345", "timestamp":1491377495217}
{"id":"scsmbstgrc", "state":"STARTED", "timestamp":1491377495210}
{"id":"scsmbstgrb", "state":"FINISHED", "timestamp":1491377495216}
...
```

In the example above, the event **scsmbstgrb** duration is 1491377495216 - 1491377495213 = 3ms

The longest event is **scsmbstgrc** (1491377495218 -1491377495210 = 8ms)

The program should:

- Take the input file path as input argument
- Flag any long events that take longer that 4ms with a column in the database called "alert".
- Write the found event details to file-based HSQLDB (http://hsqldb.org/) in the working folder
    - The application should create new table if necessary enter the following values:
        - Event id
        - Event duration
        - Type and Host if applicable
        - "alert" true is applicable

Additional points will be granted for:

- Proper use of info and debug logging
- Proper use of Object Oriented programming
- Unit tests coverage
- Multi-threaded solution
- Program that can handle very large files (gigabytes)