

Tasks and proposed solutions:

Note:

before running the Junit tests via EventLogTest.java, please ensure to set up the FILE_PATH variable

located in com.taulia.util.Constants.java

- output a new csv file where events are ordered by 'timestamp'

This get achieved by invoking the following Junit Tests:

testReadCsvlogFile

testWriteCsvEventlog

- replay the event log and output a new file for each invoice identified by 'id'.

These

files should be in JSON format and should represent invoice attributes after all the events have been replayed

This get achieved by invoking the following Junit Tests:

testWriteJsonEventlog

- test your application using unit tests to verify the above requirements are met running the Junit tests via com.taulia.EventLogTest.java.

- describe how your implementation would handle very large input files (2+ GB) or what

would need to change in order to make this happen

To speed up processing large CSV files in terms of reading/ writing some recommendations:

- Using univocity parser <https://www.univocity.com/>

- while reading CSV file wrapping the file reader with java.io.BufferedReader

- I would think handling and manipulating a large amount of data can require some sort of an in memory database

- or document database like mongoDB big, which will provide some improvement for faster transactions, Multi user concurrency

Sample Output CSV report:

| operation | timestamp | type | id | Content |
|-----------|-----------|------|----|---------|
|-----------|-----------|------|----|---------|

| | | | | |
|---------------|----------------------|---------|------|---|
| CREATE | 2018-07-23T11:02:00Z | INVOICE | INV2 | { 'id': 'INV2', 'amount': '80.00', 'currency': 'USD', 'status': 'DRAFT' } |
| CREATE | 2018-07-23T11:09:34Z | INVOICE | INV1 | { 'id': 'INV1', 'amount': '120.00', 'currency': 'USD', 'items': 2, 'status': 'DRAFT' } |
| UPDATE | 2018-07-23T11:15:10Z | INVOICE | INV1 | { 'items': 3 } |
| UPDATE | 2018-07-23T11:20:34Z | INVOICE | INV1 | { 'amount': '124.00', 'items': 4 } |
| UPDATE | 2018-07-23T11:28:34Z | INVOICE | INV1 | { 'amount': '124.40', 'status': 'NEW' } |
| DELETE | 2018-07-23T12:00:00Z | INVOICE | INV2 | |

| | | | | |
|---------------|----------------------|---------|------|--|
| CREATE | 2018-07-24T01:02:17Z | INVOICE | INV2 | { 'id': 'INV2', 'amount': '88.00', 'currency': 'USD', 'items': 5, 'status': 'DRAFT' } |
|---------------|----------------------|---------|------|--|

Sample Output JSON report:

```
{
  "INV2": [{
    "operation": "CREATE",
    "timestamp": "2018-07-23T11:02:00Z",
    "content": "{\n'id': 'INV2',\n'amount': '80.00',\n'currency': 'USD',\n'status':
'DRAFT'\n}",
    "type": "INVOICE",
    "id": "INV2"
  }, {
    "operation": "DELETE",
    "timestamp": "2018-07-23T12:00:00Z",
    "content": "",
    "type": "INVOICE",
    "id": "INV2"
  }, {
    "operation": "CREATE",
    "timestamp": "2018-07-24T01:02:17Z",
    "content": "{\n'id': 'INV2',\n'amount': '88.00',\n'currency': 'USD',\n'items':
5,\n'status': 'DRAFT'\n}",
    "type": "INVOICE",
    "id": "INV2"
  }
  ]
}
```