

Data Processing Test

Submission

Must be completed in Java.

External libraries are only allowed for Testing (Unit Test, Integration Test, etc).
Implementation has to be done with the tools provided by Java.

Your README should explain how to run your tests, and how to run the program.

Before submitting:

1. Check that given the input provided here, you can produce the same output,
2. You have written all the tests needed.
3. Your README clearly shows how we can run your tests and run your program.

There are some implementation aspects that we will take into account when we review the test:

1. Easily extensible with new events.
2. Code readability.
3. Automated Tests.

Background

FunG is a gaming company with a portfolio of fun games.

Each game emits telemetry data, so that user interactions can be tracked.

Objective

You are given a directory containing logfiles from all FunG games (see the `input` directory for an example).

Your task is to write a Java program to process these logfiles and produce 1 csv file for each tracking event.

Your program should take 2 arguments:

- `--input` which is a path to the directory containing logfiles. Assume all logfiles are directly in this path (no sub directories). Throw an exception if this doesn't exist.
- `--output` - a directory where the output files (1 for each tracking event and 1 for the invalid records) can be found. Throw an exception if this already exists.

The input data is pipe `|` separated. It is safe to assume there are no pipe characters in any of the fields so dont worry about any escaping of this character.

When processing the logfiles, hash the user ID field using sha-256.

Write the data out as comma separated data with enclosing double quotes. It is safe to assume that the input data does not contain double quotes so no need to worry about escaping. The output file name is the name of the tracking event. There is no need to write the tracking event name in each row as this is redundant.

Additionally, write any invalid records to a special invalid.log file.
Invalid records are:

- Reference a game that doesn't exist
- Reference a tracking event that doesn't exist
- Have more fields than expected
- Have less fields than expected
- Have wrong type data

Example

Input file	Input row	Validation	Output file	
input/1.log	gameinstallation jane 1 1587583122	Valid	output/gameinstallation.csv	"81f8f6dd
input/1.log	gameinstallation jane 1587583122	Invalid - too few fields	output/invalid.log	gameinsta
input/1.log	gameinstallation jane 1 1 1587583122	Invalid - too many fields	output/invalid.log	gameinsta
input/1.log	gamesetup jane 1 1587583122	Invalid - unknown event	output/invalid.log	gamesetup
input/1.log	gamepurchase jane 1 1 Free Life 1587583122	Valid	output/gamepurchase.csv	"81f8f6dd Life","158
input/2.log	gamepurchase jane 4 Skip Level Booster 1587583122	Invalid - unknown game	output/invalid.log	gamepurch
input/2.log	gamestart jane 2 1587583122	Valid	output/gamestart.csv	"81f8f6dd
input/2.log	gamestart jane 2 abcdefg	Invalid - wrong data type	output/invalid.log	gamestart
input/2.log	gamestart jane 1 1587583122	Valid	output/gamestart.csv	"81f8f6dd

Tracking Event Definitions

The following 3 tracking events can currently be emitted by FunG games. When designing your program, make sure its easy to define new tracking events in the future.

gameinstallation

This event contains the following fields:

- Event Name (String)
- User ID (String)
- Game ID (Number)
- Installation Date Time (Number)

For example to represent that user1 installed "Flying Sweets" at 22 April 2020 19:18:42 UTC, the following record is recorded in a log:

```
gameinstallation|user1|1|1587583122
```

gamestart

This event contains the following fields:

- Event Name (String)
- User ID (String)
- Game ID (Number)
- Start Date Time (Number)

For example to represent that user1 started playing "Circus Builder" at 22 April 2020 19:18:42 UTC, the following record is recorded in a log:

```
gamestart|user1|2|1587583122
```

gamepurchase

This event contains the following fields:

- Event Name (String)
- User ID (String)
- Game ID (Number)
- Product Description (String)
- Purchase Date Time (Number)

For example to represent that user1 purchased "Skip Level Booster" in game "Zoom Racer" at 22 April 2020 19:18:42 UTC, the following record is recorded in a log:

```
gamepurchase|user1|3|Skip Level Booster|1587583122
```

FunG Games

FunG currently as 3 games. Make sure we could easily introduce new games in the future.

Game ID	Game Name
1	Flying Sweets
2	Circus Builder
3	Zoom Racer