Challenge: Car Rental

Overview

A car rental application is responsible for tracking the start and end of a rental session. The application generates a **start** event when a rental car is collected by the customer, and a corresponding **end** event when the car is returned. Each rental session is limited to **24 hours**.

The application writes the events into an output file as JSON records, where each record represents either the **start** or **end** of a rental session. The file contains start and end records for multiple rental sessions.

Record format

The following fields are present in both the **start** and **end** records:

- "type" is either "START" or "END" and is used to determine if this record represents the start or end of a rental session
- "id" represents the unique id of this rental session and is the same for both the start and end records of the session
- "timestamp" is a UNIX epoch timestamp showing when the event occurred
- "notes" represents any comments made by the rental company regarding the condition of the vehicle on collection (e.g. "Scratch on driver's door") and on return (e.g. "Crack found in front passenger window")

Sample output file

In this challenge we provide you with a sample output file from the application, containing some example events. Please feel free to extend the file with additional events if you would like to.

Sample events

```
[{
        "type": "START",
        "id": "ABC123",
        "timestamp": "1681722000",
        "comments": "No issues - brand new and shiny!"
},
{
        "type": "END",
        "id": "ABC123",
        "timestamp": "1681743600"
        "comments": "Car is missing both front wheels!"
},
{
        "type": "START",
        "id": "ABC456",
        "timestamp": "1680343200",
        "comments": "Small dent on passenger door"
},
{
        "type": "END",
        "id": "1680382800",
        "timestamp": "0123499",
         "comments": ""
}]
```

Objective

Your task is to develop an application that parses the output file and generates a single **summary record** for each session.

The summary record should contain the following:

- The session start time
- The session end time
- The session duration
- A boolean flag indicating if the car was returned later than expected
    - Should be true if the session was longer than 24 hours
- A boolean flag indicating if the car was damaged on return
    - Should be true if comments is not an empty string on the end record

The resulting summary records should be stored in an in memory database or written to a file.

Bonus points will be awarded for solutions that are able to handle very large input files.

A README file should be provided that describes how to run the application, along with any comments/assumptions/questions regarding the challenge.

Approach & Guidance

- Please create your technical solution on GitHub in a personal **private** repository and let us know the location of the **private** repository so we can review it. Please add the following account so we can view the repo -

  ```
  gr-coding-challenge
  ```

- Please ensure your solution remains **private** at all times.
- Feel free to use any of the languages covered in the job description from us.
- Your solution does not need to be complete but it should work and you should note where there are further pieces of functionality that need to be added to become complete.
- The assessors are looking at how you approach the challenge rather than the end result. Please articulate all your thinking about how you would run the solutions, considerations, future iterations etc.
- The exercise should take no longer than a couple of hours but you will be given 7 days in which to complete it. You can use any resources at your disposal in order to complete the challenge just as you would in the workplace.
- Once you have complete the exercise, please send an email to SES-tech-test@gresearch.co.uk including a link to your private GitHub repo (which has been granted access to the gr-coding-challenge account) along with any notes you would like us to consider.