Lemonade
Data Engineering Challenge

Candidate Brief

Thank you for taking the time and effort to become a Lemonade Maker.

This challenge is an opportunity for you to get a better understanding of the day-to-day role in Lemonade's Engineering Team and to show us what you've got :)

It is also an opportunity for us to get to know another side of you so feel free and get creative! There is more than one way (there are tons!) to successfully complete this task.
There is no such thing as 'too many details'! Get into the specifics, we like this kinda shit ;)

Feel free to reach out with any questions you may have.
Good luck!

The Challenge

At Lemonade, we're moving fast and working with lots of data sources & data ingestion layers that are streamed into our data-lake\warehouse.

We came with a once in a lifetime solution to this problem which is to create a small dedicated data engine for this. It allows us to manage the data sources and ingestion layers magically and easily ;-)

Assumptions

- Let's assume there's a defined inbound folder that the files are landing in.
- The order of the flies landing is not promised to be on their chronological order - they can be published retroactively. It will be ordered in most cases but you cannot assume it.
- One of our data sources is external vendor data, let's assume this data arrives as files in Json format.

Challenge Description

You are asked to write a process that checks for the arrival of new files and load them into their corresponding tables in the DB.

Additional Requirements

- There are 2 types of files in the folder: (see appendix)

Vehicle_events:

i. The file name will be vehicles_events_[time].json

ii. These files will hold streaming events that are sent from the cars that we insure.

iii. Each file can hold one or more events.

Vehicles_status

i. The file name will be vehicles_status_[time].json

ii. This files will hold the latest status of each vehicle

- The assignment should also include the creation script of the tables.
- Your solution should also produce a daily summary table (see required columns below) based on the ingested events data.

vehicle_id	day	last_event_time	last_event_type

- **Important**: document everything you do & explain why you chose to do what you did. Feel free to contact us if there are any questions or things that aren't clear enough.

Notes

- Please use Python
- Please add a **readme** with clear execution instructions (solution will be tested in macos environment).
- Solutions will be measured by code structure and by their ability to handle at scale
- In case you made any assumptions, please state them so that we can better understand the implementation
- We know there's a limited time to spend on this assignment. Please state what would you do differently if you had additional time to work on it
- Any extra functionality is a bonus :-)

You're welcome to approach me for any questions you may have

Good luck!

Appendix - Event examples

vehicle_events file content sample:

```
  "vehicles_events": [
    {
      "vehicle_id": "ebab5f787798416fb2b8afc1340d7a4e",
      "event_time": "2022-06-05T21:02:34.546Z",
      "event_source": "mobile",
      "event_type": "start_drive",
      "event_value": "-79.9074173,9.3560572",
      "event_extra_data:": {
        "note": "the device is working properly",
        "boot_time": 12
      }
    },
    {
      "vehicle_id": "ebae3f787798416fb2b8afc1340d7a6d",
      "event_time": "2022-06-06T00:02:34.546Z",
      "event_source": "device",
      "event_type": "door_open",
      "event_value": "driver_door",
      "event_extra_data:": {
        "note": "a rare event, that is usually caused by
accident",
        "emergency_call": true
      }
```

```
    }
  ]
```

vehicle_status file content sample:

```
  "vehicle_status": [
    {
      "vehicle_id": "ebab5f787798416fb2b8afc1340d7a4e",
      "report_time": "2022-05-05T21:02:34.546Z",
      "status_source": "mobile",
      "status": "driving",
    },
    {
      "vehicle_id": "ebae3f787798416fb2b8afc1340d7a6d",
      "report_time": "2022-05-06T00:02:34.546Z",
      "status_source": "device",
      "status": "accident",
    },
    {
      "vehicle_id": "qbae3f787798416fb2b8afc1340ddf19",
      "report_time": "2022-05-09T00:02:34.546Z",
      "status_source": "police",
      "status": "garage",
    }
  ]
```