

Data processing exercise

What we're looking for

- Don't spend too much time. We respect your time and don't expect you to spend more than an hour or two on the exercise.
- A clean solution that handles the basics of the problem. There's no need to over-engineer.
- Clear descriptions of the choices made and why. If you have to make reasonable assumptions, please call them out.
- Code that is easy to follow and is structured according to best practices. For example, a Java solution would follow the standard Maven project layout.

Scenario

A client (e.g. a bank) sends us data whenever their customers modify their account information. For example, they may tell us when a customer changes her email address on file. The client sends this data as JSON records to a REST API endpoint, examples of which are below. Assume the account ID field is unique.

```
{"account_id": 1121345, "event_date": "2017-08-24", "account_standing": "G", "account_information": {"first_name": "John", "last_name": "Doe", "date_of_birth": "1986-08-18", "address": {"street_number": "123", "street_name": "Main Street", "city": "Centerville", "state": "CA", "zip_code": "91111"}, "email_address": "john_doe@gmail.com"}}
```

```
{"account_id": 1454581, "event_date": "2018-01-09", "account_standing": "B", "account_information": {"first_name": "Jane", "last_name": "Smith", "date_of_birth": "1975-09-09", "address": {"street_number": "345", "street_name": "Oak Drive", "unit_number": "12A", "city": "Mount Pleasant", "state": "CA", "zip_code": "90010"}, "email_address": "jane_smith@yahoo.com"}}
```

Your task is to write code to process these records to create a view with the most up-to-date information on each account. That is, we would like to query the most recent PII based on the account ID. The view can be persisted however you like (CSV, JSON, SQLite DB, etc).

Feel free to provide code in any language you're comfortable with. We'd like to see the source code and **three unit tests** to run independently for verification. You may write the unit tests however you'd like, testing whatever you'd like. Code should be uploaded to a **code sharing site** (i.e. GitHub or Bitbucket), and the link submitted to us. Name the repository whatever you'd like; we just need the link.

Follow-up questions

Please add answers to these follow-up questions in a README file inside your code repository.

1. How does your solution handle malformed or corrupt data?
2. Is your solution optimized for query latency or throughput?
3. What would you do differently if the client doesn't send the account ID?
4. If the view gets very large (can no longer fit into memory), how can we modify it to ensure we're still able to look up examples?