

# Interview assignment - Data processing

Design a micro service that processes standardized data records from multiple sources, supports aggregation and querying functions and feeds downstream services with relevant information.

## Instructions

The intended time for this assignment is a maximum of 2 hours. Document your solution in any way you are comfortable with, but be prepared to present it during the interview. The implementation can be in any programming language or even pseudo code. During the interview you will be asked to walk us through it.

You may choose any tech stack to support the service. Consider quality, scalability and performance. Document the choices and assumptions you make in your design and why.

## Requirements

1. Consume input from several services. The service should handle about 100,000 messages per hour efficiently. Implement idempotency to prevent duplicate processing, ensuring each record is processed exactly once. The data structure for incoming records is as follows:

None

```
{
  "recordId": string,
  "time": Datetime,
  "sourceId": string,
  "destinationId": string,
  "type": string ["positive"|"negative"],
  "value": Decimal,
  "unit": string,
  "reference": string,
}
```

2. Respond to queries for aggregation. The query and the response should support the

following.

1. Start and end time as optional filters.
2. The type of record (**positive** or **negative**) as optional filter.
3. Grouping by destination id. The response should include all matching records and a summarized total value per group.
3. Emit messages to be consumed by the notification service. There should be one message for every record processed. Each message should contain the processed record and a summary of any previous ones for the same destination id and reference.
4. Emit messages to be consumed by alerting service when a record's value is above a configurable threshold.

## Service overview

This diagram shows the intended place for the transactions service in the larger system.

