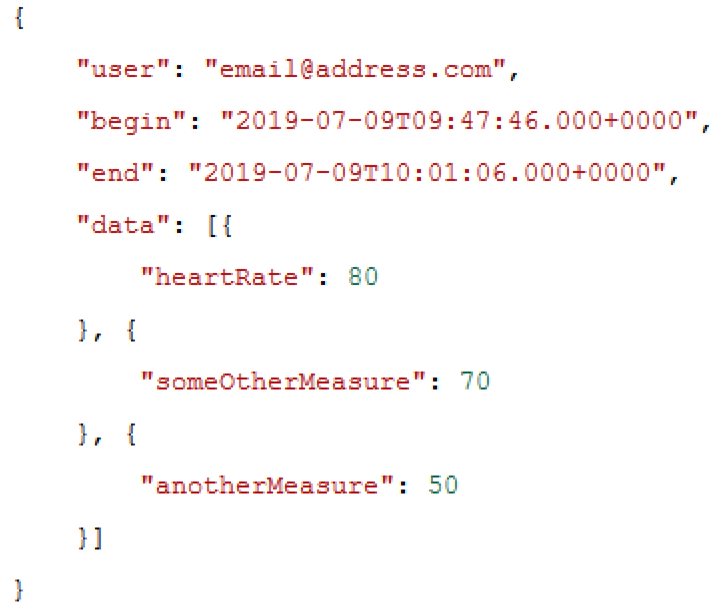
Backend Developer
Technical Test

Coding assignment

In this test, you have to code a **Spring Boot** based backend that will receive data trough an API and process data through a pipeline. Advances of the steps in the pipeline will be made available via a Websocket. Data will be stored in a relational database (in memory H2 is fine for the test)

Data model: You will receive the *session* data in JSON format with the following format:



```
{
    "user": "email@address.com",
    "begin": "2019-07-09T09:47:46.000+0000",
    "end": "2019-07-09T10:01:06.000+0000",
    "data": [{
        "heartRate": 80
    }, {
        "someOtherMeasure": 70
    }, {
        "anotherMeasure": 50
    }]
}
```

Where the table data can contain an arbitrary number of key/value (String,Double) pairs.

User is defined with the following attributes: email, first name, last name.

Web Service and Processing url:/api/receive

as soon as the web service receives the session, the session is stored in the DB with its *state* attribute set to *RECEIVED*.

Using asynchronous **Spring Events** (*ApplicationEventPublisher, ApplicationListener*) the session will be transmitted to the first stage of the pipeline. The first stage will do nothing but wait a random time between 1000 and 2000 milliseconds. The state of the session will then be set to STAGE_1. This change must be reflected in the database.

Again using events, the session will be sent to the second stage. The second stage will do nothing but wait a random time between 1000 and 2000 milliseconds. The state of the session will then be set to STAGE_2. This change must be reflected in the database.

Then the session will be transmitted to the final stage. . The second stage will do nothing but wait a random time between 1000 and 2000 milliseconds. The state of the session will then be set to PROCESSED. This change must be reflected in the database.

WebSocket

To follow the advances of the session in the pipeline, a websocket will be used to display in real time the advances of the pipeline processing.

Expected Result

To test the application, one will send a session through an API client (Insomnia, Postman) and will open a webpage in the application to follow the pipeline processing. The expected output in the browser is as follows:

TIMESTAMP – Received session ID XX with data ….

TIMESTAMP – Session ID XX Entered stage STAGE_1

TIMESTAMP – Session ID XX Entered stage STAGE_2

TIMESTAMP – Session ID XX is processed.

Delivery: the source code will be made available on github.