Employee- & Event-Service Coding Challenge

Goal

Your goal is to implement two services: ***employee-service*** and ***event-service***.

Employee-Service

As its name suggests, this service is responsible for handling the employees of a company. The application must expose a REST API. It should contain endpoints to:

1. Create an employee with the following properties
    a. Uuid (generated automatically)
    b. E-mail
    c. Full name (first and last name)
    d. Birthday (format YYYY-MM-DD)
    e. List of hobbies (for example, "soccer", "music", etc)
2. Get a specific employee by uuid (response in JSON Object format)
3. Update an employee
4. Delete an employee

Whenever an employee is created, updated or deleted, an event related to this action must be pushed in some message broker (i.e, Kafka, RabbitMq, etc). This event will be listened by the event-service.

Restriction

The email field is unique, i.e. 2 employees cannot have the same email.

Event-Service

This service is responsible for listening the events generated by employee-service and save them in its own database. The event-service must also expose a REST API that contains just one endpoint to:

1. Get all events related to a specific employee in ascending order, i.e, oldest event must appear first. Response in JSON Array format.

Bonus

(+) Use Swagger (https://swagger.io) to expose employee-service and event-service endpoints;
(+) Add authentication to access create, update and delete employee-service endpoints;
(+) Run the message broker and databases you've chosen to store the employee's data and employee event's data in Docker containers (instead of in-memory database).

Notes

- Include some unit and integration tests;
- Commit all your development using Git and share your project on GitHub;
- Include a README file explaining how to build/run your application and any comments you think it is important.