

## Exercise 6: Sagas and Stateful Resilience Patterns & Assignment 1

Deadline: 21.04.2024; 23:59 CEST

---

### Project Abstract

In the sixth week's exercise on Event-driven and Process-oriented Architectures we will investigate the implementation of stateful resilience patterns and Sagas in the context of workflow engines.

### ① Labs

In the practical part of the lecture we ask you to work on the following labs to investigate the implementation of different stateful resilience patterns and Sagas:

- Lab08 – Flowing Retail REST<sup>1</sup>: This lab shows different implementations of stateful resilience patterns using Java, Netflix/Hystrix<sup>2</sup> and the Camunda 7 BPM Platform. Try to run the project and study the behavior and implementation of the different patterns. Note that there is also an exemplary adaptation of some of these processes for the Camunda 8 (Zeebe) platform<sup>3</sup>. You can run these using the Camunda Cloud credentials in Spring Boot provided on Canvas or using the Docker compose file for the local Zeebe infrastructure (docker-compose-kafka-zeebe-infra.yml).
- Lab09 – Trip Booking Saga<sup>4</sup>: This project shows the implementation of a (distributed) transactional saga using the Camunda BPM Platform, which interacts with different services. Try to run the project and study the behavior and implementation of the trip booking process.

### ② Suggestions for Software Project

For your software application we suggest that you implement one or several of the following stateful resilience patterns:

- Stateful retry
- Human intervention
- Outbox and/or Saga patterns

Hint: There is no need for including transactions. It is ok to implement transactional sagas based on state management and eventual consistency.

### Software Project Expectations

The labs in Part 1 of this sheet are meant to be done during class and are **not graded**. In Part 2 we expect you to prepare a report on the project-related implementations that you have done in your group.

### ③ Assignment 1 Hand-in

By the deadline indicated on top of this sheet your group should hand in via the **Exercise 6** assignment module on Canvas:

---

<sup>1</sup><https://github.com/scs-edpo/lab04-flowing-retail/tree/master/rest>

<sup>2</sup><https://github.com/Netflix/Hystrix>

<sup>3</sup><https://github.com/scs-edpo/lab04-flowing-retail/tree/master/rest/java/payment-zeebe>

<sup>4</sup><https://github.com/scs-edpo/lab09-trip-booking-saga>

A zip file with your presentation slides from April 20 and a PDF describing your software application including:

- a general description of the project;
- explicit references to the concepts of the lecture and exercises and how they are covered by the project<sup>5</sup>;
- a discussion of trade-offs and important architectural decisions whenever relevant (can be complemented with ADRs);
- diagrams explaining your implementation at a conceptual level;
- any additional results and insights you have gained while working and experimenting with Kafka, Camunda and others (e.g., load tests, partitioning, impact of services not available);
- a link to a *release* version of your software application for Assignment 1 within your Git repository and additional descriptions/links related to your implementation if necessary;
- a clear indication of who in the team was responsible for which parts;
- reflections and lessons learned.

#### ④ Assignment 1 Presentations

On April 18, we will allocate a slot of approx. 25 minutes per group for a presentation of the projects including conceptual aspects as indicated for the Assignment 1 hand-in as well as a short live demo of your prototype(s) and Q&A.

We expect every member of the group to present parts of the project and to be able to answer questions related to the project.

---

<sup>5</sup>Ensure to cover the course contents more broadly rather than to focus on a single or very few aspects.