University of St.Gallen

School of Computer Science

Event-driven and Process-oriented Architectures, FS2023
B. Weber, R. Seiger, A. Abbad-Andaloussi
firstname.lastname@unisg.ch

Exercise 2: Kafka with Spring

Deadline: 07.03.2023; 23:59 CET

Project Abstract
In the second week's exercise on Event-driven and Process-oriented Architectures you will learn about using Apache Kafka together with the Spring[1] framework.

(1) Labs

In the practical part of the lecture we ask you to work on the following labs to get started with Spring and Kafka:

- Lab03Part1 – Spring for Apache Kafka[2]: This lab shows how to use the Spring framework in combination with Kafka. In class we ask you to try out this lab on your own machine and extend it with additional producers and consumers as well as different configurations. In addition, we suggest to split up the project in two individual, standalone services: 1) service(s) containing the producers and 2) service(s) containing the consumers of messages.
- Lab03Part2 – Extending the Eye-tracking Use-case with Spring for Apache Kafka[3]: This lab extends Lab03Part1 with the eye-tracking use case introduced in Lab02Part2[4]. In particular, we adapt the existing source-code to implement the eye tracking and clicks tracking producers and consumers using Kafka and Apache Spring. In class we ask you to try out this lab on your own machine and examine the parts of code differing from Lab03Part1.
- Lab04 – Flowing Retail[5]: In this lab we ask you to have a closer look at the alternative implementation of the Flowing Retail project based on choreography (take a look at the "checkout"[6] and "choreography-alternative"[7] modules). Study the implementation and investigate how the message/event exchange among services works. In class we ask you to discuss the following aspects regarding the architectural decisions per group. The aspects cover:
  - Architectural style,
  - Service granularities (i.e., bounded contexts),
  - Communication style,
  - Message flow,
  - additional error and change scenarios and their implications.

  We suggest to prepare ADRs[8] regarding your discussions and present them in class.

(2) Suggestions for Project Portfolio

Having seen how to create producers and consumers using Spring Kafka in Labs 03 Part1 and Part2, you could extend the *Flowing Retail* example from Lab04 or think about your own small project with the goal to implement at least one of the EDA patterns from Lecture 2:

[1] https://docs.spring.io/spring-kafka/reference/html/
[2] https://github.com/scs-edpo/lab03Part1-kafka-spring
[3] https://github.com/scs-edpo/lab03Part2-kafka-spring-Eye-tracking
[4] https://github.com/scs-edpo/lab02Part2-kafka-EyeTracking
[5] https://github.com/scs-edpo/lab04-flowing-retail
[6] https://github.com/scs-edpo/lab04-flowing-retail/tree/master/kafka/java/checkout
[7] https://github.com/scs-edpo/lab04-flowing-retail/tree/master/kafka/java/choreography-alternative
[8] https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions.html

comments/likes via API query every X seconds

- Event notification: add a notification service for the flowing retail example which sends emails to inform customers about relevant progress in the process.
- Event-carried state transfer: extend the checkout service such that the checkout service is aware of the available stock. This requires to keep a copy of the data at the checkout and to keep this data updated (e.g., using one of the eventual consistency pattern like event-based synchronization).
- Add new error scenarios to the Flowing Retail project.

for all videos of 1 channel

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 (portfolio)-related implementations that you have done in your group.

Hand-in Instructions

The report and implementation for the project-related Part 2 will **be graded** as part of the project portfolio for your group to be handed in for the first part of the lecture until after the semester break. An intermediate hand-in of the results from Exercise 2 via Canvas is expected together with the results from Exercise 1 by the deadline indicated on the top of this sheet. Please submit a PDF file documenting your work on the project-related part of Exercises 1 and 2 via Canvas. Include GitHub links in case there is already relevant source code. Each group member **must explicitly indicate** which part she/he/they has/have been working on. Please approach the tutors for individual feedback regarding your submission for this assignment.