

## Fachhochschul-Studiengang Wirtschaftsinformatik

### Klausur aus der Lehrveranstaltung

### Servicebasierte Systeme

Daniel Resanovic

### Hauptklausur (1. Termin)

28.2.2022

PRAXIS

### Praxis

Ein **Unternehmen** hat einen klar definierten Mitarbeiter Onboarding Prozess. D.h. bei jeder neuen Anstellung bzw. Abgang wird dieser ausgeführt.

**Employee Service** ist zuständig für die MA Verwaltung und **Workflow Service** für den Onboarding Prozess. **Employee Service** schickt alle MA via **Event Bus Service** zu **Workflow Service** für die weitere Bearbeitung im Onboarding Prozess.

Das Datenmodel soll wie folgt aussehen.

Employee Objekt

=====

Employee ID: Zufälliges ID

EmployeeName: String

EmployeePosition: Integer (1-Engineer, 2-Support, 3-Verwaltung, 4-Rest)

=====

Event Objekt

=====

EventType: String (z.B. createEmployee, deleteEmployee)

Employee ID: Zufälliges ID

EmployeeName: String

EmployeePosition: Integer (1-Engineer, 2-Support, 3-Verwaltung, 4-Rest)

=====

## Aufgaben

1. Entwickeln Sie einen **Employee Service** mit REST-API (URL-Pfad: */package*), welcher folgende Funktionen umfasst:
  - a. [15%] **POST /:positionID/employee/**: ein neuer MA wird angestellt.
    1. PositionID steht für die Job Position und kann folgende Werte haben (1-Engineer, 2-Support, 3-Verwaltung, 4-Rest). Wenn ein MA erstellt wird, soll er/sie in der **employees** Variable gespeichert sein (wir simulieren eigene DB im Service). Danach wird ein Event (siehe oben die Struktur) an den **Event Bus** weitergeleitet. **POST /events/**
    2. Falls der neue MA eine falsche Position bekommt, wird ein HTTP-417-Statuscode retourniert.
  - b. [10%] **DELETE / employee /<employee\_id>**:  
Entfernt den MA aus dem Unternehmen (aus der **employees** Variable) und retourniert die Daten des MA.
2. Entwickeln Sie für den MA Workflow einen **Event Bus** und einen **Workflow Service** Service mit REST-API (URL-Pfad: */workflow*), welcher folgende Funktionen umfasst:
  - a. [10%] **POST /events/**: **Event Bus Service** um die Pakete vom **Employee Service** an den **Workflow Service** weiterzuleiten. Benutzen sie dafür axios und die Async-Aufrufe wie in den Übungen gelernt
  - b. [5%] **GET / workflow /<mf\_employee\_id>**: Liefert Daten zum angegebenen MA retour.
  - c. [10%] **POST /mf\_employee/**: ein neuer MA wird vom **Event Bus** an die **Workflow Service** weitergeleitet. Dieser wird in eine Locale Variable **mf\_employees** gespeichert. Diese Liste ist äquivalent zu den **employees** Liste in dem **Employee Service**. Die Datenstruktur für den MA ist in beiden Services äquivalent.

## Client/Tests

Client Tests werden mit dem Postman ausgeführt.

Hinweis: Die Punktezahl der Tests stellt nicht das finale Klausurergebnis dar!

Core Applikation ist im zip File „**employeeworkflow**“. Diese nutzen, um die Applikation zu implementieren. Um alle Pakete zu installieren, benutzen sie *npm package manager*. Applikation wird mit *npm start* gestartet.