

# Personnel Administration App

## Beschreibung

Die Applikation ermöglicht dem Benutzer Personen hinzuzufügen, zu editieren und/oder zu löschen. Die Benutzeroberfläche ist intuitiv aufgebaut, als auch mit hilfreichen Tooltips versehen, sodass sie leicht zu bedienen ist.

Der allgemeine Ablauf wird anhand des folgenden Sequenzdiagramms der Funktion CreatePerson verdeutlicht (s. *create-person-sequence-diagram.png*):

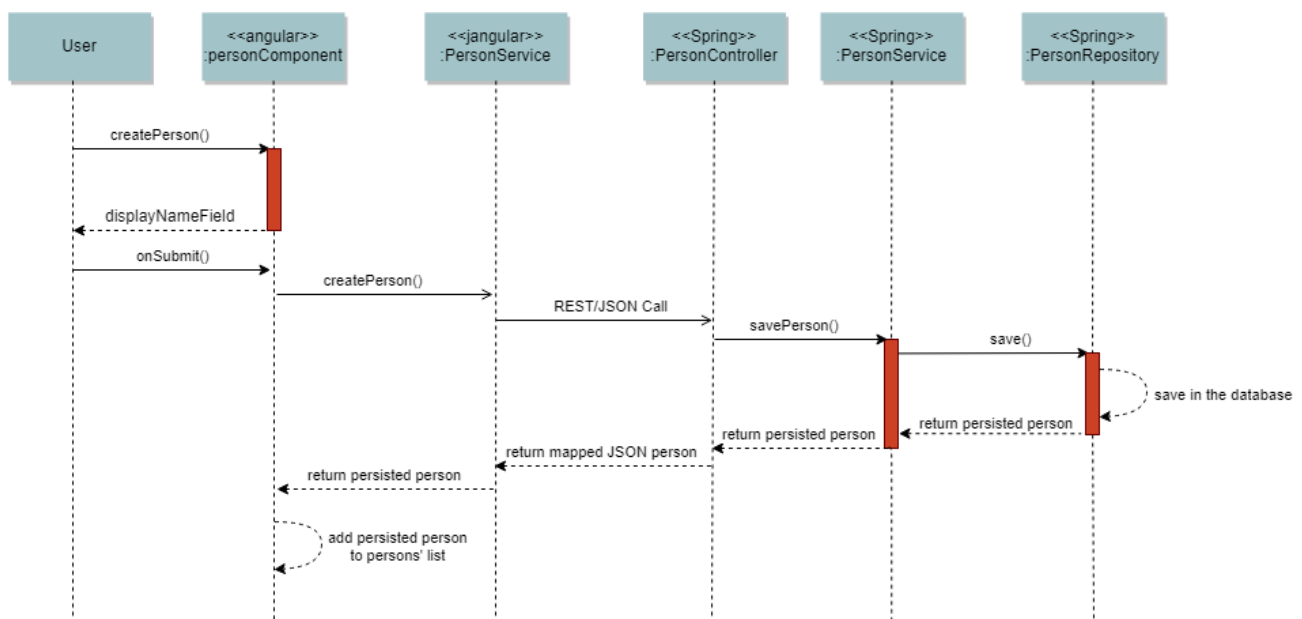


Abbildung 1: Create person sequence diagram

## Aufbau

Die WebApplikation besteht aus zwei Modulen, dem Frontend und dem Backend. Eine Übersicht des Aufbaus ist in dem Dokument *Architecture.pdf* dargestellt.

## Frontend

Frontend ist mit dem Angular Framework und der Programmiersprache TypeScript entwickelt worden. Darüber hinaus wurden für die Darstellung das Frontend-CSS-Framework Bootstrap und die Icons-Bibliothek Fontawesome hinzugefügt.

Im Wesentlichen besteht der Quellcode aus den folgenden Teilen:

## Components

Ein Komponent besteht aus einer HTML Datei für die Darstellung, einem TypeScript für die Business Logik und einer CSS Datei für das Stylen.

## Services

Die Services stellen eine Reihe von Business Logik für die Komponenten bereit. In dieser Applikation nutzt das entwickelte PersonService das HTTP Modul vom Angular, um mit der Backend-Rest-API zu kommunizieren.

## Backend

Das Backend ist mit dem Spring Framework und der Programmiersprache Java entwickelt worden. Dabei bietet das Spring Web Modul eine integrierte REST API, inklusive das Mapping zwischen Java und Json Objekten.

Der Quellcode besteht hauptsächlich aus den folgende Teilen:

## Rest Controllers

Rest Controllers stellen die REST API Funktionalität bereit. Es wird anhand des URL Pfad und der HTTP Method entschieden, welche Funktion aufgerufen wird. Diese Funktion ruft dementsprechend den geeigneten Person Service Code auf.

## Services

Die Services sammeln die Funktionalitäten, die zusammen gehören. In dieser Applikation ist der PersonService entwickelt worden, der anhand der PersonRepository mit der Datenbank kommunizieren kann. Somit jede Aktion, die mit der Bearbeitung von den Personen Daten und/oder Logik hierüber läuft.

## Repositories

Das Spring Framework bietet eine Implementierung der JPA, die im Hintergrund Hibernate nutzt. In der Applikation wurde das Interface PersonRepository definiert, das das JPA Interface vom Spring erweitert. Somit sind Standard Funktionen, wie z.B. das Speichern und Löschen der Daten, schon vorhanden. Für weitere Funktionen soll das Interface mit eindeutigen Methoden-Definitionen erweitert werden, die ein von Spring vorgegebenes Schema verfolgen. Diese Funktionen werden automatisch vom Framework implementiert.

## Entities

Hier erfolgt das Mapping zwischen den Datenbank Tabellen und den Java Klassen. Dabei wird jedes Feld der Tabelle zur gleichnamigen Klassenvariable zugeordnet. Bei einem unterschiedlichen Name muss der Benutzer eine entsprechende Annotation mit dem passenden Tabellen-Feld hinzufügen.

## Logging

Das Logging erfolgt via SLF4J mit der Logback-Implementierung. Das Logging wird in dieser Applikation erweitert, um jede Aktion des Benutzers in der Konsole auszugeben.

## Testing

Es sind bereit zwei Test Klassen definiert. Die erste Klasse testet die wichtigsten Funktionen der Applikation über die PersonService Klasse und die zweite Simuliert HTTP Anfragen und testet somit die REST API.

## Projekt Erzeugen und Starten

Der Quellcode des Projekts befindet sich unter der Git Repository:  
<https://github.com/sabbane/persons-app.git>

Es handelt sich um ein Maven Projekt, das in zwei Module aufgeteilt ist - das Backend und das Frontend. Über die *pom.xml* Datei im Hauptprojekt kann das Projekt mit dem folgenden Befehl erzeugt werden: *mvn install*

Das resultierte JAR Datei befindet sich unter dem Backend Folder *persons-app/backend/target/backend-<version>.jar*.

Die erzeugte JAR Datei kann direkt ausgeführt werden via Befehl *java -jar filename.jar* . Dabei wird ein eingebetteter Tomcat-Server sowie eine In Memory Datenbank (H2) gestartet. Somit sind keine extra Konfigurationen notwendig.

Die Applikation ist via Webbrowser unter der Adresse *http://localhost:8080* erreichbar.