Arbeitspaket - REST-Datenbank & Testing

Voraussetzungen

Installationsvoraussetzung: **JDK 18, NodeJS, Angular**

Solltet ihr es noch nicht getan haben, solltet ihr euch nun bevor ihr mit dem Arbeiten beginnt, den Backend Foliensatz ansehen, dort werden euch die Grundlagen von Springboot erklärt.
Es könnte sein, dass ihr vor dem Bearbeiten des Frontends zunächst noch den Ordner "installieren" müsst, damit dieses lauffähig ist.
Geht dazu in den Order Client und gebt in eurer Konsole/Terminal „npm install“ ein.
Dabei sollte dann der Node Package Manager alle nötigen Libraries installieren, damit Angular starten kann.

Aufgabe 1

In diesem Arbeitspaket bekommt ihr einen Springboot Server mit H2-Datenbank und eine Angular Frontend zu Verfügung gestellt.
Beginnt damit im Backend die Klassen Person.java und Adress.java zu bearbeiten, damit diese in der Datenbank gespeichert werden und folgende Eigenschaften besitzen:

Person:

1. ID, welche automatisch generiert wird.
2. Vornamen
3. Nachnamen
4. Geburtsjahr
5. Adresse, welche über einen Fremdschlüssel gespeichert wird.

Adresse:

1. ID, welche automatisch gespeichert wird
2. Stadt
3. Straße

Man soll die Möglichkeit haben im Personen nach Geburtsjahr filtern.
Man soll Personen nach der Stadt suchen können.

Bearbeitet dazu im Backend die Klassen Person, Adress, PersonController und PersonRepository.
Bearbeitet im Client die Dateien: Person.service.ts, Person-Table.component.ts, Person-Form.component.ts.

Hilfe

https://spring.io/guides/gs/accessing-data-jpa/

Aufgabe 2

Im Folgenden möchtet ihr euren Server so erweitern, dass dieser nun die folgende weitere Abfrage bearbeiten kann.

> Es soll auf Basis der vorher implementierten Geburtsjahr Filtermethode das Durchschnittsalter der gefundenen Personen berechnet werden.

Implementiert zunächst diese Methode.

Bevor ihr jedoch den Server direkt mit dieser Methode erweitert solltet ihr diese entsprechend testen.
Hierfür solltet ihr eine Testklassen schreiben, welche die entsprechende Methode testet.
Wichtig ist hierbei das Auflösen der Dependencies, da wir auf Basis von temporär erstellten, hardgecodeten Daten testen werden.

Hilfe

https://site.mockito.org/
https://junit.org/junit5/docs/current/user-guide/#writing-tests/