

AngularMaterialCRUD

In diesem Repository sollen die vollen Fähigkeiten des Backends ausgeschöpft werden. Beim Request-Assignment wurden nur die bestehenden Patienten gelesen und in einer Liste - ohne Angular Material - angezeigt.

Anforderungen an das Frontend

Im Frontend ist bereits eine `package.json` mit Angular Material vorkonfiguriert. Erstelle mittels eines Schematic zunächst eine Navigation. Darin dann eine AddressForm und erweitere diese um alle Felder um einen Patient (mit mehreren Namen, Anschriften, etc.) hinzuzufügen.

Befehl zum generieren der Navigation Schematic

```
ng generate @angular/material:navigation <component-name>
```

Befehl zum generieren der Address-Form Schematic

```
ng generate @angular/material:address-form <component-name>
```

Zum Anzeigen bestehender Patienten kannst du den Quellcode aus dem AngularRequestAssignment teilweise wiederverwenden. Diese Liste soll als Tabelle angezeigt werden. Dafür gibt es ebenfalls eine Schematic.

Befehl zum generieren der Table Schematic

```
ng generate @angular/material:table <component-name>
```

Nachdem im Backend folgendes möglich ist (bzw. sein sollte)

- Alle Patienten abfragen
- Patient mit bestimmter ID abfragen
- Patient hinzufügen
- Patient ändern
- Patient löschen

sind diese Funktionen im Frontend zu implementieren:

- Alle Patienten anzeigen (je Patient ein Bearbeiten & Löschen-Button)
 - Klickt man auf einen Patienten kommt man via einem RouterLink in die Detailansicht
- Detailansicht eines Patienten die ID wird aus der URL ausgelesen
 - Dort sollen die Daten als reiner Text (in divs oder spans) angezeigt werden
 - Mit Klick auf einen bearbeiten-Button soll die Komponente ausgetauscht werden gegen ein Formular welches alle Felder bearbeitbar macht

Anforderungen an das Backend

Damit die Anwendung unabhängig von der lokalen Umgebung (also XAMPP) ausführbar ist, ist die Datenbank in einer von beiden Varianten bereitzustellen:

- Als Docker-Container

- Die H2-Datenbank verwenden (oder eine andere In-Memory Datenbank)

Der Server ist im Backend einzufügen. Entweder das komplette Maven-Projekt oder nur die Jar-Datei.

Abgaberichtlinien

Gib in einem Unterordner deinen Server dazu, mit `mvn spring-boot:run` werde ich diesen dann ausführen um das Frontend zu unterstützen. Achte darauf dass die `.gitignore`-Datei angewandt wird um die Repository-Größe möglichst gering zu halten.

Das Frontend werde ich mit `ng serve` starten. Auch hier bitte darauf achten dass `node_modules`, `.angular` etc. nicht committed werden.

Bewertungsgrundlagen

- Backend ausführbar 10P
 - Der Befehl `mvn spring-boot:run` stürzt nicht ab
- Frontend ausführbar 10P
 - Der Befehl `ng serve` stürzt nicht ab
- Patientenliste wird angezeigt 20P
 - Ein HTTP-Request mit den Patienten kommt an und ausgewählte Spalten können in der Übersicht angezeigt werden.
- Patient kann bearbeitet werden 20P
 - Klickt man in der Übersicht auf `Edit` kommt man auf `/patient/:id`. Dort sind ganz viele bearbeitbare Textfelder
- Patient kann gelöscht werden 20P
 - Durch einen klick auf `Delete` wird ein HTTP Request ausgeführt um den Patienten mit gegebener Id aus der DB zu entfernen
- Einzelner Patient wird angezeigt 20P
 - Gleich wie Edit nur das die Felder nicht bearbeitbar sein sollten

Lade in deinem Repository eine Datei `bewertung.md` hoch in der du jedem der obigen Anforderungen (Teil)Punkte gibst. Begründe deine Punkteanzahl.

Tutorials zum Nachlesen

[Routing](#)[Schematics](#)[HTTP](#)