



Angular Aufgaben

Kleines Projekt

Zeit: ca. 2 Stunden

Datum: 02.02.2023

Author: Andy Bui

Themen

- Git anwenden
- Angular aufsetzen
- Layout & Design
- Routing kennen lernen
- Daten, Variablen und Datenaufbereitung

Materialien in dieser Übung

- NodeJS
- Visual Studio Code
- Git / Sourcetree
- Internet-Zugang
- beliebiger Browser

Übung 1

Angular 15 mit Material - Einrichtung

Lade das Projekt herunter und erstelle einen neuen Zweig mit deinen Namen:

<https://github.com/AndyNope/simple-angular-material.git>

Im Projekt ist bereits ein Layout und Angular Material Design eingebaut.

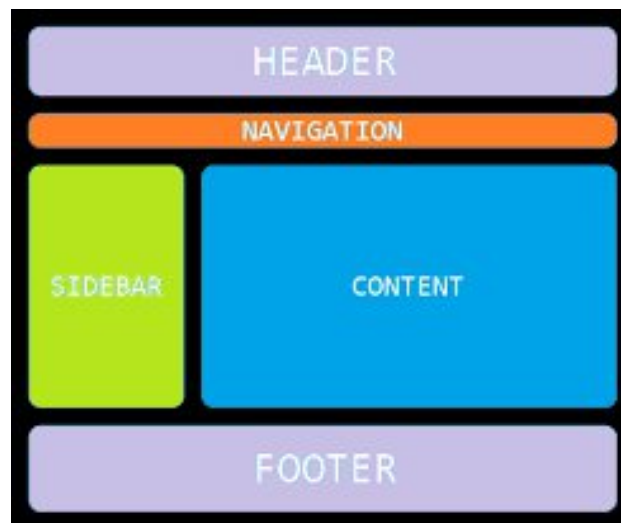
Für das Projekt fehlen jedoch noch Module, welches mit Node.js (in der Konsole) noch aufgesetzt werden muss. Angular-CLI könnte in diesem Fall noch fehlen.

Aufgabe:

Das Projekt sollte auf **localhost:4200** angezeigt werden.

Aktuelles Layout:

- Header
- Navigation
- Slidenbar
- Main-Container / Body
- Footer



Design-Werkzeuge

- CSS / SCSS
- Material-Design
- Media Query
- Bootstrap / Tailwindcss (optional)

Bootstrap oder andere Styling-Drittanbieter (optional)

Falls du dich mit andere CSS-Framework wie zum Beispiel Bootstrap oder Tailwindcss auskennst und damit arbeiten möchtest, kannst du es gerne einbauen.

Übung 2

Login-Seite

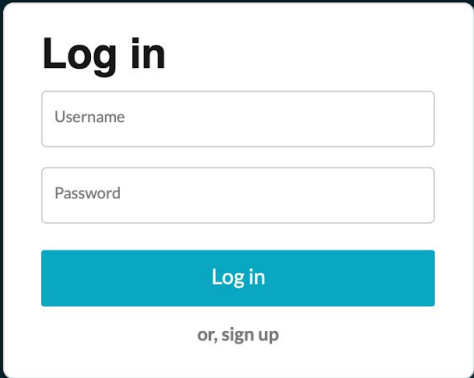
In diesem Fall haben wir für Login Logout bereits ein Beispiel eingebaut. Die ist unter localhost:4200/login erreichbar. Du kannst die Daten via Console vom Browser anschauen.

Aufgabe: Baue nun die Loginseite mit einem Login-Formular. Verwende hierzu “reactive forms” von Angular. Seite 2 sollte ohne Anmeldung nicht erreichbar sein. Wenn du nicht weiterkommst, kannst du dich gerne an mich wenden.

Hilfestellung: <https://angular.io/guide/reactive-forms>

Übung 3

Styling der Login-Seite



Log in

Username

Password

Log in

or, sign up

Aufgabe:

Jetzt kommt deine CSS-Kenntnis in Frage! Ich bitte dich darum, das Login als Box darzustellen und in der Mitte vom **Main-Container** zu platzieren.

Styles wurden hier noch nicht hinzugefügt. Hier kannst du es selbst gestalten oder Material Design (ist bereits installiert) verwenden:

Hilfestellung: <https://material.angular.io/components/categories>

Übung 4

Routing und Auth-Guard

Um dem Zugriff auf der Hauptseite zu verhindern, muss du CanActivate noch hinzufügen.

Hilfestellung: <https://angular.io/api/router/CanActivate>

Aufgabe:

Hier wurde **CanActivate** bereits im **AuthGuard-Service** eingebaut. Dies kannst du im Routing vom Projekt einbauen. CanActivate muss du noch im Routing nach dem Pfad hinzufügen. Für die Login-Kontrolle kannst du die API testen. Falls du nicht weiterkommst, melde dich bitte bei mir oder du kannst diese Aufgabe vorerst überspringen. Verwende für die Weiterleitung [RouterModule](#).

Übung 5 - Daten, Variablen und Datenaufbereitung

Aufgabe:

Rufe die API <https://api.andynope.com/clients/> via Service auf und liste den Namen auf der Startseite in die Tabelle auf. Sobald man auf einer Zeile der Tabelle klickt, sollte via Typscript ein Alert mit den kompletten Kundendaten anzeigen werden.

Falls du noch Zeit hast, kannst du den auch einen Modal/Dialog hinzufügen.

Beispiel: <https://material.angular.io/components/dialog/overview>

Daten JSON & API

API GET: <https://api.andynope.com/clients/>

Response:

```
[{
  "id": string,
  "name": string",
  "phone": string,
  "email": string,
  "address": string,
  "postalZip": string,
  "region": string,
  "country": string
}]
```

API POST: <https://api.andynope.com/login/>

Request:

```
{
  "username": "KreativMedia",
  "password": "b3E!13hy"
}
```

API GET: <https://api.andynope.com/logout/>

API GET: <https://api.andynope.com/session/>