


Aufgabenblatt 3: Webclient-Programmierung mit Angular


Ziel dieses Aufgabenblattes ist es, die in der Vorlesung besprochenen Konzepte des Angular-Frameworks zu vertiefen. Hierfür entwickeln Sie eine Homepage mit Angular. Abbildung 1 gibt vor, wie diese Homepage aussehen soll. Die weiteren Ausführungen beziehen sich jeweils auf das Beispiel. Im Kursraum finden sie ein ZIP-Archiv namens „homepage.zip“, welches ein Lösungsfragment als Startpunkt für die Entwicklung Ihrer eigenen Lösung enthält.



Willkommen auf der Homepage von Bart Simpson!

Steckbrief mehr Details... meine Links

Mein Steckbrief:

 EAT MY SHORTS, MAN!

Name: Bart Simpson

Geburtsdatum: 24.12.2003

Ort: Hollywood

Beruf: Sohn

Bildpfad: bart.jpeg

ändern

Abbildung 1: Homepage

Die zu entwickelnde Angular-Anwendung funktioniert folgendermaßen:

- Die Anwendung wird durch Eingabe von „ng serve“ in einem eigenen Terminalfenster (in Visual Studio Code: STRG+ö unter Windows) gestartet.
- Nach dem Start wird die in Abbildung 1 dargestellte Seite angezeigt (*single page web application*). Die drei Tabs („Steckbrief“, „mehr Details“, „meine Links“) sind voll funktionstüchtig, d.h. man kann zwischen den Tabs hin und her wechseln, wobei jedoch nur das erste Tab einen Inhalt hat.
- Das Steckbrief-Tab zeigt die dargestellten Inhalte zunächst an und lässt keine Bearbeitung zu, da die Formularfelder gesperrt (*disabled*) sind.

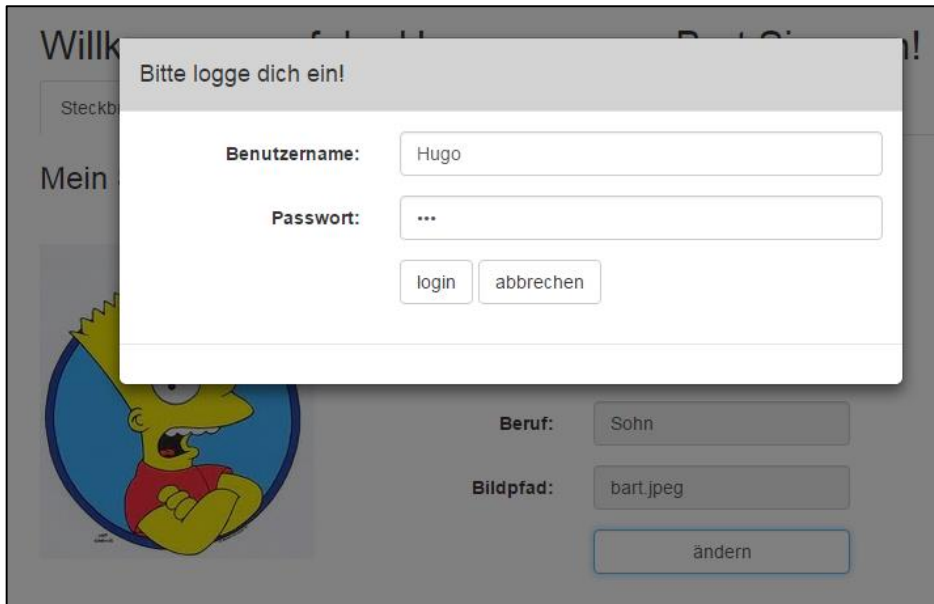


Abbildung 2: Dialog zum Eingeben von Benutzername und Passwort



Abbildung 3: Fehlermeldung bei Eingabe eines falschen Passworts

- Durch Klick auf den Button „ändern“ öffnet sich ein modaler Login-Dialog, siehe Abbildung 2. Nach Befüllen der Felder und Klicken auf den Button „login“ wird geprüft, ob „Hugo“ bzw. „123“ als Benutzername bzw. Passwort eingegeben wurden. Fehlerhafte Eingaben bewirken eine Reaktion wie in Abbildung 3 gezeigt. Der Abbrechen-Button bewirkt das Schließen des Dialogs.
- Nach erfolgreichem Login sind alle Formularfelder bearbeitbar, siehe Abbildung 4. Es wird ein Button mit der Aufschrift „ok“ angezeigt. Dieser bewirkt das Speichern der Daten nach erfolgter Eingabe. Vor dem Speichern wird geprüft, ob ein nicht-leeres Geburtsdatum eingegeben wurde. Falls nicht, wird das letzte (gültige) Geburtsdatum wiederhergestellt. Nach erfolgtem Speichern der ggf. veränderten Daten befindet sich der Dialog wieder im ursprünglichen Modus, d.h. es können keine Daten geändert werden. Entsprechend wird wie zu Beginn der „ändern“-Button angezeigt, der „ok“-Button dagegen nicht.

Willkommen auf der Homepage von Bart Simpson!

[Steckbrief](#) [mehr Details...](#) [meine Links](#)

Mein Steckbrief:



Name:

Geburtsdatum:

Ort:

Beruf:

Bildpfad:

Abbildung 4: Bearbeiten der Daten nach erfolgreichem Login

- Die Daten werden mithilfe eines Angular-Service namens „SpeicherService“ geladen und gespeichert (siehe Datei „speicherService.ts“). Zur Vereinfachung der Implementierung soll hier der „local storage“ des Browsers verwendet werden (vgl. Aufgabenblatt 2). In einer echten Anwendung, die die Daten über einen Webserver lädt und speichert, könnte stattdessen z.B. ein REST-Service (siehe Aufgabenblatt 5) aufgerufen werden. Zur Umsetzung dieser Vorgehensweise muss nur der Service, nicht jedoch der den Service nutzende Controller angepasst werden (Vorteil der Modularisierung durch Controller und Services).

Willkommen auf der Homepage von Bart Simpson!

[Steckbrief](#) [mehr Details...](#) [meine Links](#)

Mein Steckbrief:



Name:

Geburtsdatum:

Ort:

Beruf:

Bildpfad:

Abbildung 5: Fehlermeldung nach Eingabe eines ungültigen Bildpfades

- Wurde ein ungültiger Bildpfad eingegeben, so wird anstelle des Bildes eine Fehlermeldung angezeigt (vgl. Abbildung 5).

Hinweise:

- **Installation von nodeJS und npm:** Installieren Sie beides von <https://www.npmjs.com/get-npm>.
- **Installation von Angular-cli:** Öffnen Sie das homepage-Verzeichnis mit Visual Studio Code („Datei > Ordner öffnen“ im VSC-Menü). Öffnen Sie nun ein Terminal-Fenster im homepage-Verzeichnis (Menüpunkt „Anzeigen > integriertes Terminal“) und starten Sie das Kommando „**npm install -g @angular/cli@latest**“
Daraufhin wird das Command Line Interface von Angular installiert.
- **Starten der Anwendung:** Starten Sie das Kommando „**npm install**“. Es werden nun alle benötigten node-Module installiert bzw. aktualisiert. Geben Sie als nächstes „**ng serve**“ ein. Es wird daraufhin ein node-Webserver auf Port 4200 gestartet. Im Browser können Sie nun die Anwendung unter dem URL <http://localhost:4200> nutzen.
- **jQuery:**
Zum Ausblenden des Login-Dialogs nach Betätigen des „abbrechen“-Buttons wurde die jQuery-Bibliothek genutzt (s.ref <https://jquery.com/>). Informationen zur Nutzung von jQuery für modale Dialoge finden Sie unter http://www.w3schools.com/bootstrap/bootstrap_ref_js_modal.asp
- **Bootstrap** (siehe <http://www.w3schools.com/bootstrap/default.asp>):
Bootstrap bietet Webentwicklern eine Menge an Möglichkeiten, die hier genutzten Features sind:
 - CSS-Styles für die Anwendung, z.B. für die Gestaltung des Formulars für die Daten des Benutzers (siehe z.B. http://www.w3schools.com/bootstrap/bootstrap_forms.asp)
 - Umsetzung der „tabs“ mithilfe des Tab-Konzepts von Bootstrap (siehe z.B. http://www.w3schools.com/bootstrap/bootstrap_tabs_pills.asp)
 - Umsetzung des Login-Dialogs mit dem „modal“-Konzept von Bootstrap (siehe z.B. http://www.w3schools.com/bootstrap/bootstrap_modal.asp)

Vorgehensweise und Aufgabenstellung:

1. Arbeiten Sie sich in das gegebene Lösungsfragment unter „src/app“ ein und versuchen Sie, die Grundstruktur zu erkennen. Nutzen Sie dabei die in den Hinweisen gegebenen Links sowie die dort zu findenden Beispiele, um den Einsatz des jeweiligen Konzepts zu verstehen.
2. Vervollständigen Sie die Lücken im Code des Controllers (Codestellen a), b) und c)), so dass Ihre Lösung wie beschrieben funktioniert (und auch so aussieht!).

3. Vervollständigen Sie die Lücken im Code des Service (Laden und Speichern von Personen im LocalStorage).
4. Erweitern Sie den Dialog um zwei Felder, ein Benutzername und ein Passwort-Feld (eigener input-Typ!), in welchen man den Benutzernamen und das Passwort (als neue Properties der Personendaten) verändern kann. Der Login-Dialog soll entsprechend auf diese Daten (und nicht mehr auf Hugo/123) zugreifen.
5. Verstecken Sie die unter 3. eingeführten Felder in der gesperrten Ansicht, so dass man den Benutzernamen und das Passwort nur nach erfolgreichem Login sehen und verändern kann.

Abnahme:

Die Lösung ist einzeln vorzuführen und Fragen zur Funktionsweise (auf Code-Ebene) sind zu beantworten. Bitte beachten Sie: auch bereits vorgegebener Code sollte erklärt werden können!