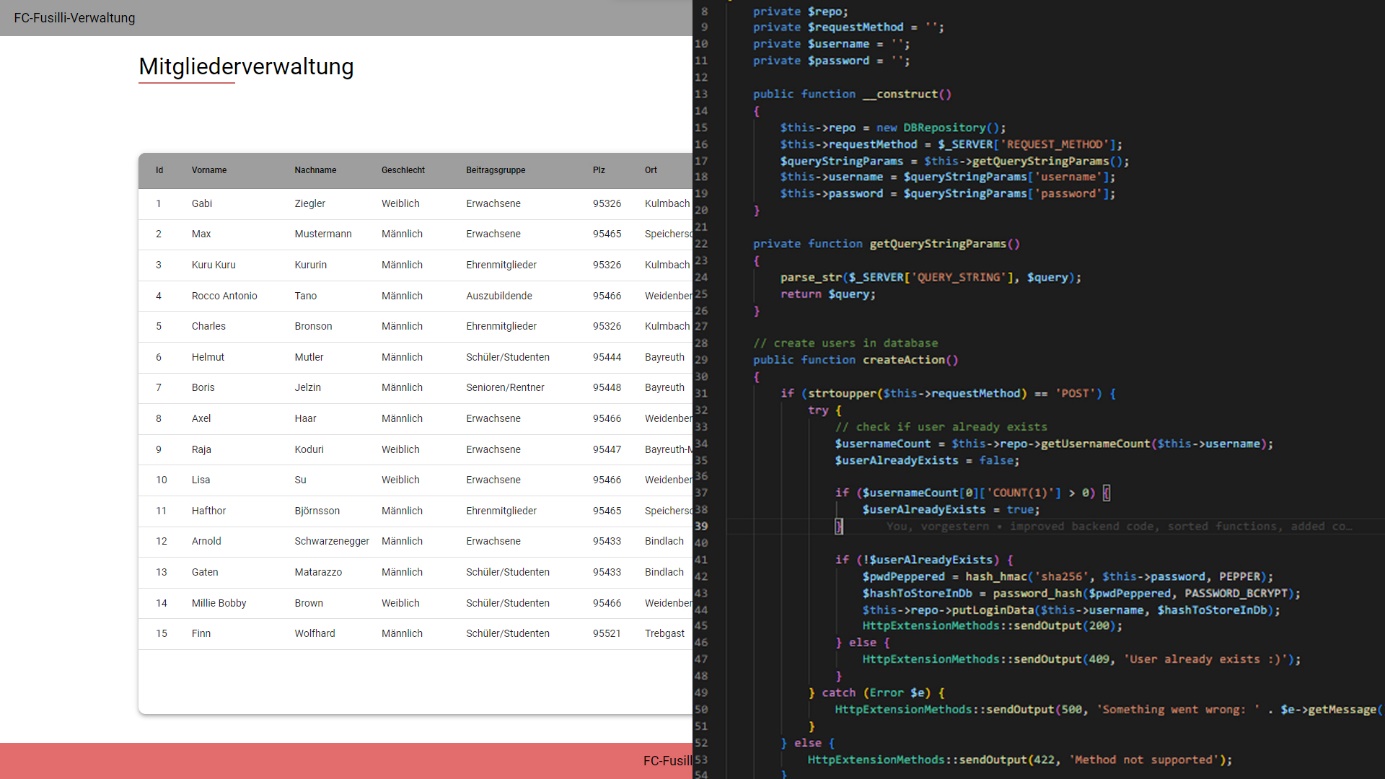
Juli - 2022

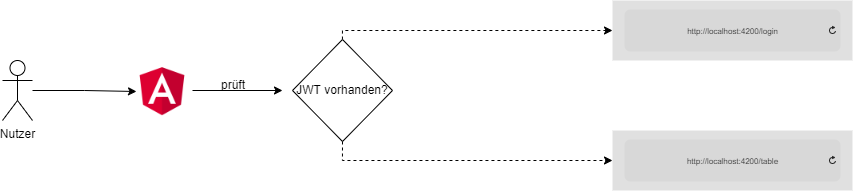
WEBBASIERTER-DATENBANKMANAGER



Gruppe 4 – Fusilli

|  |  |
| --- | --- |
| Thomas S. | @Noodle693 |
| Marco H. | @GodlesZ95 |
| Samuel S. | @Gunzli02 |
| Michael W. | @miwied |

Inhaltsverzeichnis

**Authentifizierung**

**Vorgehensweise**

Der Nutzer kann nur auf die Daten zugreifen, wenn er sich vorher mittels Log-in authentifiziert und dadurch einen JWT erhalten hat.

Der Token wird per Local-Storage in der Anwendung abgelegt und bleibt dort solange erhalten, bis er gelöscht oder die Anwendung beendet wird.

  setToken(token: string) {

    localStorage.setItem('token', token);

  }

Anschließend wird beim Aufrufen der Route geprüft, ob der User eingeloggt ist.

  isLoggedIn() {

    const usertoken = this.getToken();

    if (usertoken != null) {

      return true;

    }

    return false;

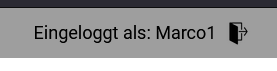
  }

Wenn kein Token vorhanden ist, wird der User wieder auf die Login-Seite weitergeleitet.

**JSON Web Token**

Neben der Authentifizierung wird der JSON Web Token außerdem dafür verwendet, den Namen aus dem Token zu extrahieren und im Frontend anzuzeigen

 ngAfterContentInit() {

    const helper = new JwtHelperService();

    var decodedToken = helper.decodeToken<any>(

      this.apiService.getToken()?.toString()

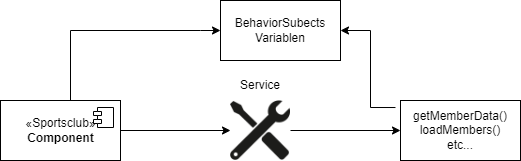
    );

    this.tokenUserName = decodedToken.userName;

  }

Mithilfe des JwtHelperService Packages kann der Token decodiert werden. Danach kann der Name ausgelesen und angezeigt werden.

**Austausch der Daten**



In Angular arbeiten wir mit Komponenten. Diese beinhalten das HTML-Gerüst und die Logik in Form von Typescript. Zusätzlich haben wir verschiedene Services, die zum einen als Repository die Daten im Backend abrufen und zum anderen dem Datenaustausch intern dienen, integriert. Im Kern stehen hier sogenannte BehaviorSubjects und Observables.

Observable, Subject und BehaviorSubject stammen aus dem RxJS Framework. Diese Klassen sind nicht Bestandteil von JavaScript und Angular. BehaviorSubject erbt von Subject und das erbt von Observable. Alle haben den gleichen Feature-Umfang wie ein Observable.

Ein Subject speichert das letzte Event, welches über das Observable verschickt wurde. Ein Zugriff auf das letzte Event ist in einem Observable nicht möglich.

Ein BehaviorSubject hat zusätzlich einen initialen Wert, der zurückgeliefert wird, wenn noch kein Event verschickt wurde.

**Validierung**

Damit die Integrität der Daten gewährleistet wird, ist es im Frontend nur möglich, valide Anfragen an das Backend zu schicken. Angular gibt uns hierbei die Möglichkeit, mithilfe der FormControl individuelle Validierungsregeln für einzelne Eingabefelder zu definieren. So können z. B. bei der PLZ lediglich Zahlen eingegeben werden.

       <mat-form-field appearance="outline" name="city">

            <mat-label>Ort</mat-label>

            <input matInput formControlName="city" placeholder="Musterstadt" />

          </mat-form-field>

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung  zipCode: new FormControl(null, [

        Validators.required,

        Validators.pattern('^[0-9]\*$'),

        Validators.minLength(4),

        Validators.maxLength(6),

      ]),