

## Teil 1 – Web mit Angular

---

### Installation

---

```
npm install -g windows-build-tools
npm install -g @angular/cli
```

### Erstellung des Grundgerüsts

---

```
ng new MugDemo --directory=01_Web --prefix=mug --skip-git --skip-tests
cd 01_Web
rmdir /Q /S node_modules
code .
```

In **package.json** unter den **dependencies** die aktuelle Version von Angular (**~7.0.0-rc.0**) durch eine ältere, mit Ionic kompatible Version ersetzen (**5.2.11**). Dasselbe für rxjs machen (**~6.3.3** zu **5.5.11**).

```
npm install
ng serve --open
```

### Installation von Semantic UI

---

```
npm install --save semantic-ui-css
```

Anpassung von **angular.json**: unter "styles" folgenden Pfad ergänzen:

```
"node_modules/semantic-ui-css/semantic.css"
```

Zum Test der Installation den Inhalt von **app.component.html** ersetzen:

```
<h1 class="ui horizontal divider header" style="padding-top: 40px; padding-bottom: 20px;">
  <i class="users icon"></i>
  Mitglieder der MUG Zentralschweiz
</h1>
<div class="ui placeholder segment" style="margin: 10px;">
  <!-- <mug-member-list></mug-member-list> -->
</div>
```

Anschliessend die Website neu starten:

```
ng serve
```

### Erstellung der Models

---

```
ng g class model/member
```

Inhalt von **model\member.ts** ersetzen:

```
export class Member {
  constructor(
    public name: string,
    public city: string,
    public profileUrl: string,
    public imageUrl: string
  ) {}
}
```

## Erstellung des Service

```
ng g service service/member-repository
```

Inhalt von **service\member-repository.service.ts** ersetzen:

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Member } from '../model/member';
import { Observable } from 'rxjs';
import 'rxjs/add/operator/map';

@Injectable()
export class MemberRepositoryService {

  constructor(private http: HttpClient) { }

  public getAll(): Observable<Member[]> {
    return new Observable<Member[]>(obs => {
      this.http.get('assets/members.json').subscribe((membersDTO: any) => {
        const members = new Array<Member>();

        for (const memberDTO of membersDTO.results) {
          const member = new Member(
            memberDTO.name,
            memberDTO.city,
            memberDTO.link,
            memberDTO.photo.photo_link);

          members.push(member);
        }

        obs.next(members);
        obs.complete();
      });
    });
  }
}
```

Anpassung von **app.module.ts**:

```
import { HttpClientModule } from '@angular/common/http';
import { MemberRepositoryService } from '../service/member-repository.service';

// ...

imports: [
  BrowserModule,
  HttpClientModule
],
providers: [
  MemberRepositoryService
],

// ...
```

Einen Output des [Webservice](#) in **assets/members.json** hinterlegen.

## Erstellung der Komponente

---

ng g component member-list

Inhalt von **member-list\member-list.component.ts** ersetzen:

```
import { Member } from '../model/member';
import { MemberRepositoryService } from '../service/member-repository.service';
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'mug-member-list',
  templateUrl: './member-list.component.html',
  styleUrls: ['./member-list.component.css']
})
export class MemberListComponent implements OnInit {
  public members: Member[];

  constructor(private membersRepository: MemberRepositoryService) {
  }

  ngOnInit() {
    this.membersRepository.getAll().subscribe(
      loadedMembers => this.members = loadedMembers
    );
  }
}
```

Inhalt von **member-list\member-list.component.html** ersetzen:

```
<div class="ui cards">
  <div class="ui raised centered card" *ngFor="let member of members">
    <div class="content">
      <img class="right floated ui avatar image" style="width: 60px; height: 60px;"
[src]="member.imageUrl">
      <a class="header" [href]="member.profileUrl">{{member.name}}</a>
      <div class="meta">{{member.city}}</div>
    </div>
  </div>
</div>
```

Die Komponente sollte von der CLI automatisch in **app.module.ts** registriert worden sein. Nun muss die Komponente aber noch in **app.component.html** aktiviert werden (Kommentare entfernen).

## Teil 2 – Desktop mit Electron

---

### Erstellung des Grundgerüsts

---

```
xcopy 01_Web 02_Desktop /E /I
cd 02_Desktop
rmdir /Q /S node_modules
npm install
npm install electron --save-dev
code .
```

### Erstellung der main.js-Datei im Root-Verzeichnis

---

Basierend auf folgendem [Tutorial](#) den Inhalt von **main.js** erstellen (Root-Verzeichnis):

```
const { app, BrowserWindow } = require('electron')

const path = require('path')
const url = require('url')

let win

function createWindow () {
  win = new BrowserWindow({ width: 800, height: 600, autoHideMenuBar: true })

  win.loadFile('dist/MugDemo/index.html');

  win.on('closed', () => {
    win = null
  })
}

app.on('ready', createWindow)

app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') {
    app.quit()
  }
})

app.on('activate', () => {
  if (win === null) {
    createWindow()
  }
})
```

### Anpassung der index.html-Datei

---

```
<base href="./">
```

### Anpassung der package.json-Datei

---

```
"main": "main.js",                // Auf Hauptebene
"electron": "ng build && electron .", // Unter "scripts"
```

### Ausführung der Elektron-Anwendung

---

```
npm run electron
```

## Teil 3 – Mobile mit Ionic

---

### Erstellung des Grundgerüsts

---

```
ionic start 03_Mobile blank
cd 03_Mobile
code .
ionic serve --lab
```

### Integration der bestehenden Lösung

---

Selektiv Teile der bestehenden Web-App in die Mobile-App kopieren:

src /app/model/*	→	src/app/model/*
src/app/service/*	→	src/app/service/*
src/app/member-list/*	→	src/app/member-list/*
src/assets/members.json	→	src/assets/members.json

### Kopierte Elemente korrekt einbinden

---

Anpassung der **app.module.ts** wie folgt:

```
import { HttpClientModule } from '@angular/common/http';
import { MemberRepositoryService } from '../service/member-repository.service';
import { MemberListComponent } from '../member-list/member-list.component';

// ...

declarations: [
  MyApp,
  HomePage,
  MemberListComponent
],

// ...

imports: [
  BrowserModule,
  HttpClientModule,
  IonicModule.forRoot(MyApp)
],

// ...

providers: [
  StatusBar,
  SplashScreen,
  {provide: ErrorHandler, useClass: IonicErrorHandler},
  MemberRepositoryService
]
```

In **member-list/member-list-component.ts** die Referenz auf das CSS-File entfernen:

```
@Component({
  selector: 'mug-member-list',
  templateUrl: '../member-list.component.html'
})
```

Einbau der MemberList-Komponente in **pages/home/home.html**:

```
<ion-content padding>
  <mug-member-list></mug-member-list>
</ion-content>
```

Ausführen der aktuellen Lösung: noch OHNE UI-Komponenten von Ionic:

```
ionic serve --lab
```

## Verwendung der UI-Komponenten von Ionic

---

Inhalt von **member-list/member-list-component.html** ersetzen:

```
<ion-card *ngFor="let member of members" style="margin-bottom: 20px !important;">
  <ion-item>
    <ion-avatar item-end>
      <img [src]="member.imageUrl">
    </ion-avatar>
    <h2>{{member.name}}</h2>
    <p>{{member.city}}</p>
  </ion-item>
</ion-card>
```

## Deployment auf Android-Smartphone

---

```
ionic cordova platform add android
ionic cordova run android
```