

Ionic 7 - Element aus Liste auswählen und weiterleiten mit Hilfe der ID für die Detailseite

Inhalt:

1. Seite „mitarbdetail“ erstellen
2. apiService http.get erstellen
3. Auswahl weiterleiten (mitarbeiter.page.html)
4. Auswahl mit GET entgegennehmen (mitarbdetail.page.ts)
5. Detailseite erstellen (mitarbdetail.page.html)
6. Das API anpassen
7. Andere Möglichkeit der Weiterleitung
8. alle Seiten als Bildschirmausdruck



1)Erstelle eine neue Seite in den „pages“

```
3> ionic g page pages/mitarbdetail
```

Öffne app-routing.modules.ts

Nun wurde die Seite geschaffen. Die Route wird in der Datei „app-routing.modules.ts“ automatisch erstellt. Da wir aber nicht auf die Detailseite als solches surfen wollen, nutzt uns das so nichts. Diese Seite „mitarbdetail“ wird nur mit der einen ausgewählten Information derjenigen ID ausgefüllt, die vorher in der Liste der „mitarbeiter“ gewählt wurde.

Daher verändere den „path“ um den Zusatz „/:id“ als Platzhalter.

```
46 | {  
47 |   path: mitarbdetail/:id',  
48 |   loadChildren: () => import('./pages/mitarbdetail/mitarbdetail.mod  
49 | },
```

Nur zur Info:

Dieser Platzhalter wird dann **später** verwendet (mit Hilfe einer GET-Funktion in details.page.ts:

```
18 |   ngOnInit() {  
19 |     const id = this.route.snapshot.paramMap.get('id');
```

2)Erstelle das Service für die ID:

Öffne apiService.ts

Kopiere „getMitarbeiter“ und ändere diese Funktion auf „getDetails“ – es leitet ebenfalls auf die API mit Hilfe einer GET-Methode:

- In der Klammer setze die „id“ auf „any“.
- Damit der richtige Manager aufgerufen werden kann, wird am Ende der Schrägstrich und dann nach dem schließenden Anführungszeichen mit einem Plus die „id“ angehängt.

```
ex.php/mitarbdetail/'+id);
```

```
50 getMitarbeiter()
51 {
52   return this.http.get('https://ionic.digbizmistelbach.info/api_einkaufsliste/public/index.php/mitarbeiter');
53 }
54
55 getMitarbDetail(id: any) {
56   return this.http.get('https://ionic.digbizmistelbach.info/api_einkaufsliste/public/index.php/mitarbdetail/'+id);
57 }
58
```

3)Auswahl weiterleiten

Öffne die „mitarbeiter.page.html“

Erstelle eine an die anderen Seiten angepasste „toolbar“:

back-Button linksbündig

```
<ion-button slot="start">
  <ion-back-button></ion-back-button>
</ion-button>
```

```
Go to component
1 <ion-header [translucent]="true">
2   <ion-toolbar color="success">
3     <ion-title>Details des Mitarbeiters</ion-title>
4     <ion-button slot="start">
5       <ion-back-button></ion-back-button>
6     </ion-button>
7   </ion-toolbar>
8 </ion-header>
9
10 <ion-content [fullscreen]="true">
```

In der Liste muss nun der betreffend geklickte Eintrag, bzw. dessen ID, weitergeleitet werden.

Man könnte einen Button einfügen bzw. wie hier, den ganzen „item“ klickbar machen:

Mit einem Klick auf den Button in der Liste soll es funktionieren:

- Dafür muss der Button klickbar sein – mittels „button“
- Und einen Pfad aufweisen: routerLink

- Und bei Verwendung von routerLink auch die „private route: activatedRoute“ in constructor und import anlegen -> in der TS der „mitarbdetail.page.ts“ – siehe nächster Punkt.
- Die Route lautet „/mitarbdetail/“ an die man die „userid“ aus der Datenbank anhängt - in der **Datenbank** steht „userid“. Dies geschieht mit der Doppelten geschwungenen Klammer, die in IONIC für die Interpolation steht und sehr einfach zu verwenden ist.

```
routerLink="/mitarbdetail/{{mitarbeiter.iduser}}">
```

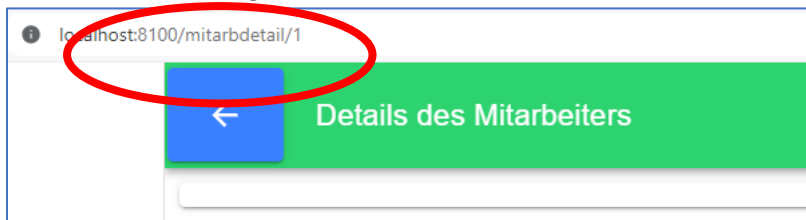
routerLink="/mitarbdetail/{{mitarbeiter.iduser}}"

```

15 <ion-label><h5>Startzeit Schicht: {{mitarbeiter.startzeit }}</h5></ion-label><br>
16 <ion-button color="primary" routerLink="/mitarbdetail/{{mitarbeiter.iduser}}">Detail</ion-button>
17 </ion-row>
18 </ion-grid>

```

Das führt zur Weitergabe der ID in der URL



INFO:

man kann statt einem extra Button zum Weiterleiten auch die ganze Zeile „klickbar“ machen:

```

7 <ion-content>
8 <ion-list>
9 <ion-item lines="insert" button *ngFor="let manager of managers" routerLink="/detail/{{manager.userid}}">
10 <ion-label><h5>Vorname:{{manager.vorname}}</h5></ion-label><br>

```

4)Auswahl mit GET annehmen

Öffne die mitarbdetail.page.ts

Im constructor

- die Route erstellen/aktiveren:
Damit soll die Information über die aktuelle ID gegeben werden. Diese ist die jeweilige des geklickten Elements aus der Liste.
- Bezug auf die ApiService

Beachte, dass auch der Import erfolgt, am besten automatisiert.

```

1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3 import { ApiService } from 'src/app/api.service';
4
5 @Component({
6   selector: 'app-mitarbdetail',
7   templateUrl: './mitarbdetail.page.html',
8   styleUrls: ['./mitarbdetail.page.scss'],
9 })
10 export class MitarbdetailPage implements OnInit {
11
12   constructor(
13     private route: ActivatedRoute,
14     private apiService: ApiService
15   ) {}

```

Im Aufbau der Datei soll diese nun gleich benutzt werden, daher schreibe das in die „ngOnInit“: man könnte es auch mittels Funktion machen und die Funktion in der ngOnInit aufrufen.

```

13   ngOnInit() {
14     const id = this.route.snapshot.paramMap.get('id');
15   }

```

const id = this.route.snapshot.paramMap.get('id');
console.log(id);

```

18   ngOnInit() {
19     const id = this.route.snapshot.paramMap.get('id');
20     console.log(id);
21   }

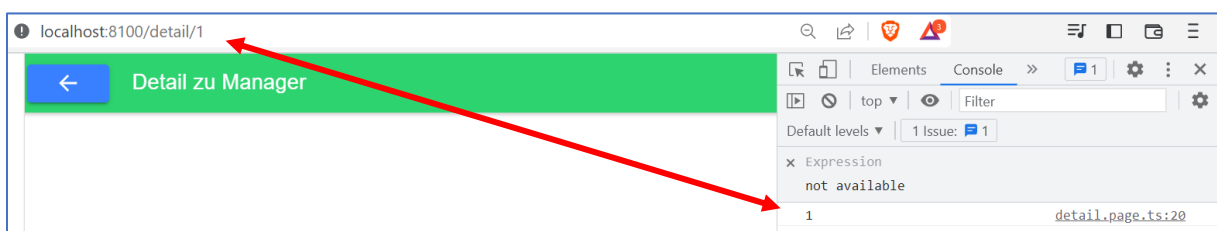
```

console.log(id);

Konsole aufrufen mit der Taste F12:

zwischendurch nutzt die Ausgabe zur Kontrolle

Damit kann man die Ausgabe und Korrektheit überprüfen, wenn die Zahl in der Konsole ausgegeben wird.



Darunter – hier Zeile 21 und 25: - zur Verbindung zum apiService:

```

18   ngOnInit() {
19     const id = this.route.snapshot.paramMap.get('id');
20     // console.log(id);
21     this.apiService.getMitarbDetail(id).subscribe((response: any) => {
22       this.mitarbdetails = response;
23     });
24   }
25 }
26

```

ngOnInit() {

```

    const id = this.route.snapshot.paramMap.get('id');
    // console.log(id);
    this.apiService.getMitarbDetail(id).subscribe((response: any) => {
    this.mitarbdetails = response;
    });
}

```

Aber auch im export oben:

```

10 export class MitarbdetailPage implements OnInit {
11     mitarbdetails: any;
12
13     constructor(

```

Beachte: diese Klasse „mitarbdetails“ wird dann in der *ngFor-Schleife der HTML benötigt.

5)Detailseite erstellen

Im content sollen die Detailinfos aus der Datenbank, mit Hilfe der API, angeführt werden.

```

10 <ion-content [fullscreen]="true">
11     <ion-card color="light">
12         <ion-list>
13             <ion-item lines="insert" *ngFor="let item of mitarbdetails">
14                 <ion-label><h5>Benutzer:{{item.benutzername}}</h5></ion-label><br>
15                 <ion-label><h5>E-Mail:{{item.mail}}</h5></ion-label><br>
16                 <ion-label><h5>Passwort: {{item.passwort }}</h5></ion-label><br>
17                 <ion-label><h5>Eintrittsdatum: {{item.datum }}</h5></ion-label><br>
18                 <ion-label><h5>Schichtbeginn: {{item.startzeit }}</h5></ion-label><br>
19             </ion-item>
20         </ion-list>
21     </ion-card>
22 </ion-content>

```

```

<ion-card color="light">
  <ion-list>
    <ion-item lines="insert" *ngFor="let item of mitarbdetails">
      <ion-label><h5>Benutzer:{{item.benutzername}}</h5></ion-label><br>
      <ion-label><h5>E-Mail:{{item.mail}}</h5></ion-label><br>
      <ion-label><h5>Passwort: {{item.passwort }}</h5></ion-label><br>
      <ion-label><h5>Eintrittsdatum: {{item.datum }}</h5></ion-label><br>
      <ion-label><h5>Schichtbeginn: {{item.startzeit }}</h5></ion-label><br>
    </ion-item>
  </ion-list>
</ion-card>

```

6)Kommunikation mit der API

Die Kommunikation erfolgt mit dem API, die auf dem Server oder localhost liegt.

API index.php

Wie die „manager“ aber am Beginn:

```
154 $app->get('/mitarbdetail/{id}', function( Request $request, Response $response){
155     $id = $request->getAttribute('id');
156     $sql = "SELECT * FROM user WHERE iduser = '$id' ";
157
158     try {
159         $db = new db();
```

1. Angehängt wird {id}
2. \$id = \$request...
3. Ein WHERE in der SQL

```
$app->get('/mitarbdetail/{id}', function( Request $request, Response $response){
    $id = $request->getAttribute('id');
    $sql = "SELECT * FROM user WHERE iduser = '$id' ";
```

```
try {
    $db = new db();
    $db = $db->connect();
    $stmt = $db->query( $sql );
    $user = $stmt->fetchAll( PDO::FETCH_OBJ );
    $db = null; // clear db object

    // print out the result as json format
    echo json_encode( $user );

} catch( PDOException $e ) {
    // show error message as Json format
    echo '{"error": {"msg": ' . $e->getMessage() . '}}';
}
});
```

```

154 $app->get('/mitarbdetail/{id}', function( Request $request, Response $response){
155     $id = $request->getAttribute('id');
156     $sql = "SELECT * FROM user WHERE iduser = '$id' ";
157
158     try {
159         $db = new db();
160         $db = $db->connect();
161         $stmt = $db->query( $sql );
162         $user = $stmt->fetchAll( PDO::FETCH_OBJ );
163         $db = null; // clear db object
164
165         // print out the result as json format
166         echo json_encode( $user );
167
168     } catch( PDOException $e ) {
169         // show error message as json format
170         echo '{"error": {"msg": ' . $e->getMessage() . '}}';
171     }
172 });
173

```

7) andere Infos und alle Seiten als Code-Bild

Weiterleitung aus dem Klick in der Liste **auch so möglich:**

Statt „routerlink“ etwas umständlicher auch mit der click-Funktion in der HTML und der passenden Funktion in der TS.

```

7   <ion-content>
8   <ion-list>
9   <ion-item lines="insert" button *ngFor="let manager of managers"
10  <(click)="detailzeigen(manager.userid)">
11  <ion-label><h5>Vorname:{{manager.vorname}}</h5></ion-label><br>

```

TS:

```

29  detailzeigen(id: string){
30  this.nav.navigateForward('detail/'+id);
31  }
32
33  }

```

Oder mit Buttons

- Für das Editieren
- löschen

8)Codes

mitarbdetail.page.html

```
Go to component
1  <ion-header [translucent]="true">
2  <ion-toolbar color="success">
3    <ion-title>Details des Mitarbeiters</ion-title>
4  <ion-button slot="start">
5    <ion-back-button></ion-back-button>
6  </ion-button>
7  </ion-toolbar>
8  </ion-header>
9
10 <ion-content [fullscreen]="true">
11 <ion-card color="light">
12 <ion-list>
13 <ion-item lines="insert" *ngFor="let item of mitarbdetails">
14   <ion-label><h5>Benutzer:{{item.benutzername}}</h5></ion-label><br>
15   <ion-label><h5>E-Mail:{{item.mail}}</h5></ion-label><br>
16   <ion-label><h5>Passwort: {{item.passwort }}</h5></ion-label><br>
17   <ion-label><h5>Eintrittsdatum: {{item.datum }}</h5></ion-label><br>
18   <ion-label><h5>Schichtbeginn: {{item.startzeit }}</h5></ion-label><br>
19
20   <ion-button color="primary" routerLink="/mitarbedit/{{item.iduser}}">
21     <ion-icon name="create"></ion-icon>
22   </ion-button>
23   <ion-button color="danger" routerLink="/mitarbdetail/{{item.iduser}}">
24     <ion-icon name="trash"></ion-icon>
25   </ion-button>
26 </ion-item>
27 </ion-list>
28 </ion-card>
29 </ion-content>
30
```

mitarbdetails.ts:

```
1  import { Component, OnInit } from '@angular/core';
2  import { ActivatedRoute } from '@angular/router';
3  import { ApiService } from 'src/app/api.service';
4
5  @Component({
6    selector: 'app-mitarbdetail',
7    templateUrl: './mitarbdetail.page.html',
8    styleUrls: ['./mitarbdetail.page.scss'],
9  })
10 export class MitarbdetailPage implements OnInit {
11   mitarbdetails: any;
12
13   constructor(
14     private route: ActivatedRoute,
15     private apiService: ApiService
16   ) {}
17
18   ngOnInit() {
19     const id = this.route.snapshot.paramMap.get('id');
20     // console.log(id);
21     this.apiService.getMitarbDetail(id).subscribe((response: any) => {
22       this.mitarbdetails = response;
23     });
24   }
25 }
```

mitarbeiter.ts

```
1  import { Component, OnInit } from '@angular/core';
2  import { ApiService } from 'src/app/api.service';
3
4  @Component({
5    selector: 'app-mitarbeiter',
6    templateUrl: './mitarbeiter.page.html',
7    styleUrls: ['./mitarbeiter.page.scss'],
8  })
9  export class MitarbeiterPage implements OnInit {
10   mitarbeiters: any[] | undefined;
11
12   constructor(
13     private apiService: ApiService
14   ) {
15     this.getMitarbeiter();
16   }
17
18   ngOnInit() { }
19
20   getMitarbeiter(){
21     this.apiService.getMitarbeiter().subscribe((response: any) => {
22       this.mitarbeiters = response;
23     });
24   };
25 }
```

manager.html

```
1  <ion-header [translucent]="true">
2    <ion-toolbar color="success">
3      <ion-title>Das sind unsere Mitarbeiter</ion-title>
4    </ion-toolbar>
5  </ion-header>
6
7  <ion-content [fullscreen]="true">
8    <ion-list>
9      <ion-item lines="insert" *ngFor="let mitarbeiter of mitarbeiters">
10       <ion-grid>
11
12         <ion-list>
13           <ion-label><h5><b>Benutzername: {{mitarbeiter.benutzername}}</b></h5></ion-label><br>
14         </ion-list>
15         <ion-list>
16           <ion-label><h5>E-Mail: {{mitarbeiter.mail}}</h5></ion-label><br>
17         </ion-list>
18         <ion-list>
19           <ion-label><h5>Eintrittsdatum: {{mitarbeiter.datum}}</h5></ion-label><br>
20         </ion-list>
21         <ion-list>
22           <ion-label><h5>Startzeit Schicht: {{mitarbeiter.startzeit}}</h5></ion-label><br>
23         </ion-list>
24         <ion-list>
25           <ion-button color="primary" routerLink="/mitarbdetail/{{mitarbeiter.iduser}}">Detail</ion-button>
26         </ion-list>
27
28       </ion-grid>
29     </ion-item>
30   </ion-list>
31 </ion-content>
```

API:

```

128 $app->get('/mitarbeiter', function( Request $request, Response $response){
129     // echo 'Benutzer';
130
131     $sql = "SELECT * FROM user";
132
133     try {
134         // Get DB Object
135         $db = new db();
136
137         $db = $db->connect();
138         $stmt = $db->query( $sql );
139         $user = $stmt->fetchAll( PDO::FETCH_OBJ );
140         $db = null; // clear db object
141         // print out the result as json format
142         echo json_encode( $user );
143         //print_r($user);
144
145     } catch( PDOException $e ) {
146
147         // show error message as Json format
148         echo '{"error": {"msg": ' . $e->getMessage() . '}}';
149     }
150
151 });
152
153

```

```

154 $app->get('/mitarbdetail/{id}', function( Request $request, Response $response){
155     $id = $request->getAttribute('id');
156     $sql = "SELECT * FROM user WHERE iduser = '$id' ";
157
158     try {
159         $db = new db();
160         $db = $db->connect();
161         $stmt = $db->query( $sql );
162         $user = $stmt->fetchAll( PDO::FETCH_OBJ );
163         $db = null; // clear db object
164
165         // print out the result as json format
166         echo json_encode( $user );
167
168     } catch( PDOException $e ) {
169         // show error message as Json format
170         echo '{"error": {"msg": ' . $e->getMessage() . '}}';
171     }
172
173 });
174

```

Quelle: Simon Grimm https://www.youtube.com/watch?v=y_vwf15eADs

Auch gut: aber von firebase

Teil 19: https://www.youtube.com/watch?v=sN9Oc_E-TQ (Detailseite anlegen)

Teil 20: <https://www.youtube.com/watch?v=ZcwB0CR9xvg> (ID nicht aus DB, Weiterleiten nach Buttonklick auf Detail mit Nummer in URL)

Ionic und crud: <https://www.youtube.com/watch?v=bqiHfIBh8Xk>

<https://ionic.io/ionicons>