

Exercise (Instructions): Single Page Applications Part 1

Objectives and Outcomes

In this exercise you will continue to develop the Angular application as a single page application, integrating the various components. At the end of this exercise you will be able to:

- Leverage the Angular router to enable the development of single page applications
- Provide a way of navigating among various views using the Angular router support.

Integrating the Contact Component

- Open the `contact.component.html` file and update its contents as follows:

```

1  <div class="container"
2    fxLayout="column"
3    fxLayoutGap="10px">
4
5    <div fxFlex>
6      <div>
7        <h3>Contact Us</h3>
8        <hr>
9      </div>
10   </div>
11
12   <div fxFlex>
13     <h3>Location Information</h3>
14     <div fxLayout="column" fxLayout.gt-sm="row">
15       <div fxFlex="50" fxFlexOffset="20px">
16         <h4>Our Address</h4>
17         <address>
18           121, Clear Water Bay Road<br>
19           Clear Water Bay, Kowloon<br>
20           HONG KONG<br>
21           <i class="fa fa-phone"></i>: +852 1234 5678<br>
22           <i class="fa fa-fax"></i>: +852 8765 4321<br>
23           <i class="fa fa-envelope"></i>:
24             <a href="mailto:confusion@food.net">confusion@food.net
25               </a>
26         </address>
27         <p></p>
28         <div>
29           <a md-raised-button href="tel:+85212345678"><i class="fa fa
30             -phone"></i> Call</a>
31           <a md-raised-button><i class="fa fa-skype"></i> Skype</a>
32           <a md-raised-button href="mailto:confusion@food.net"><i
33             class="fa fa-envelope-o"></i> Email</a>
34         </div>
35       </div>
36     </div>
37     <div fxFlex="40">
38       <h4>Map of our Location</h4>
39     </div>
40   </div>
41 </div>

```

- Update the routes.ts file to add in the configuration for the contact component by adding in the following to the routes constant:

```

1  { path: 'contactus',    component: ContactComponent }

```

- Update the header.component.html file to activate the contact link:

```

1  <a md-button routerLink="/contactus"><span class="fa fa-address-card fa-lg"
2    ></span> Contact</a>

```

Updating the Home Component

- First update dish.ts file to update the dish class as follows:

```

1 import { Comment } from './comment';
2
3 export class Dish {
4   id: number;
5   name: string;
6   image: string;
7   category: string;
8   label: string;
9   price: string;
10  featured: boolean;
11  description: string;
12  comments: Comment[];
13 }

```

- Now that we added the id and featured property to dish, update the DISHES constant in dishes.ts to add an id to each element, and also add a featured property to each dish as follows:

```

1 [
2   {
3     id: 0,
4     name: 'Uthappizza',
5     . . .
6     featured: true,
7
8     . . .
9   },
10  {
11    id: 1,
12    name: 'Zucchipakoda',
13    . . .
14    featured: false,
15
16    . . .
17  },
18  {
19    id: 2,
20    name: 'Vadonut',
21    . . .
22    featured: false,
23
24    . . .
25  },
26  {
27    id: 3,
28    name: 'ElaiCheese Cake',
29    . . .
30    featured: false,
31
32    . . .
33  }
34 ];

```

- Now update the dish service to return a specific dish, and a featured dish as follows:

```

1 getDish(id: number): Dish {
2   return DISHES.filter((dish) => (dish.id === id))[0];
3 }
4
5 getFeaturedDish(): Dish {
6   return DISHES.filter((dish) => dish.featured)[0];
7 }

```

- Next add promotion.ts file to the shared folder and update its contents as follows:

```

1 export class Promotion {
2   id: number;
3   name: string;
4   image: string;
5   label: string;
6   price: string;
7   featured: boolean;
8   description: string;
9 }

```

- Then, add promotions.ts file to the shared folder and update its contents as follows:

```

1 import { Promotion } from './promotion';
2
3 export const PROMOTIONS: Promotion[] = [
4   {
5     id: 0,
6     name: 'Weekend Grand Buffet',
7     image: '/assets/images/buffet.png',
8     label: 'New',
9     price: '19.99',
10    featured: true,
11    description: 'Featuring mouthwatering combinations with a choice of five
        different salads, six enticing appetizers, six main entrees and five
        choicest desserts. Free flowing bubbly and soft drinks. All for just $19
        .99 per person'
12  }
13 ];

```

- Now add a new service named "promotion" to the services folder.
- Then, update the promotion.service.ts file as follows:

```

1 import { Injectable } from '@angular/core';
2 import { Promotion } from '../shared/promotion';
3 import { PROMOTIONS } from '../shared/promotions';
4
5 @Injectable()
6 export class PromotionService {
7
8   constructor() { }
9
10  getPromotions(): Promotion[] {
11    return PROMOTIONS;
12  }
13
14  getPromotion(id: number): Promotion {
15    return PROMOTIONS.filter((promo) => (promo.id === id))[0];
16  }
17
18  getFeaturedPromotion(): Promotion {
19    return PROMOTIONS.filter((promotion) => promotion.featured)[0];
20  }
21 }

```

- Import the promotion service into the AppModule and add the service to the providers.
- Update the home.component.html file as follows to add two cards to show the featured dish and promotion in the home page:

```

1 <div class="container"
2   fxLayout="row"
3   fxLayout.sm="column"
4   fxLayout.xs="column"
5   fxLayoutAlign.gt-md="space-around center"
6   fxLayoutGap="10px">
7
8 <md-card *ngIf="dish" fxFlex>
9   <md-card-header>
10    <div md-card-avatar></div>
11    <md-card-title>
12      <h3>{{dish.name | uppercase}}</h3>
13    </md-card-title>
14  </md-card-header>
15  <img md-card-image src={{dish.image}} alt={{dish.name}}>
16  <md-card-content>
17    <p>{{dish.description}}
18  </p>
19  </md-card-content>
20 </md-card>
21
22 <md-card *ngIf="promotion" fxFlex>
23   <md-card-header>
24    <div md-card-avatar></div>
25    <md-card-title>
26      <h3>{{promotion.name | uppercase}}</h3>
27    </md-card-title>
28  </md-card-header>
29  <img md-card-image src={{promotion.image}} alt={{promotion.name}}>
30  <md-card-content>
31    <p>{{promotion.description}}
32  </p>
33  </md-card-content>
34 </md-card>
35
36 </div>

```

- Now update the home.component.ts file to fetch and provide the featured dish and promotion for the view:

```

1 import { Component, OnInit } from '@angular/core';
2
3 import { Dish } from '../shared/dish';
4 import { DishService } from '../services/dish.service';
5 import { Promotion } from '../shared/promotion';
6 import { PromotionService } from '../services/promotion.service';
7
8 @Component({
9   selector: 'app-home',
10  templateUrl: './home.component.html',
11  styleUrls: ['./home.component.scss']
12 })
13 export class HomeComponent implements OnInit {
14
15   dish: Dish;
16   promotion: Promotion;
17
18   constructor(private dishservice: DishService,
19     private promotionservice: PromotionService) { }
20
21   ngOnInit() {
22     this.dish = this.dishservice.getFeaturedDish();
23     this.promotion = this.promotionservice.getFeaturedPromotion();
24   }
25
26 }

```

Highlighting the Current Component Link in the Toolbar

- Update each of the links in the toolbar with the following addition to the link:

```

1 <a . . . routerLinkActive="active"><span . . . ></span> Home</a>

```

- Now add the following scss class to header.component.scss:

```

1 . . .
2 $background-moredark: #4527A0;
3 . . .
4 .active {
5   background: $background-moredark;
6 }

```

- Save all the changes and do a Git commit with the message "Single Page Applications Part 1".

Conclusions

In this exercise you developed the Angular application further by integrating the components into a single page application.