Exercise (Instructions): Angular Routing Basics

Objectives and Outcomes

In this exercise you will be adding three new additional components to your application. You will then configure a basic router using the Angular router module to enable your application to navigate among the components. At the end of this exercise you will be able to:

- Configure your application with multiple components whose views can be shown one at a time and navigate among them

- Configure a basic router using the Angular routing module to enable the navigation among the views.

Adding Components

- Add three new components to your Angular application as follows:

```
1  ng generate component about
2  ng generate component home
3  ng generate component contact
```

Add a Router Module

- Add a new module named app-routing to your application as follows. This will create a new module file named *app-routing.module.ts* in the *app-routing* folder.

```
1  ng generate module app-routing
```

- Next create a new file named *routes.ts* in the *app-routing* folder and update it as follows:

```
1    import { Routes } from '@angular/router';
2
3    import { MenuComponent } from '../menu/menu.component';
4    import { DishdetailComponent } from '../dishdetail/dishdetail.component';
5    import { HomeComponent } from '../home/home.component';
6    import { AboutComponent } from '../about/about.component';
7    import { ContactComponent } from '../contact/contact.component';
8
9    export const routes: Routes = [
10     { path: 'home',   component: HomeComponent },
11     { path: 'menu',      component: MenuComponent },
12     { path: '', redirectTo: '/home', pathMatch: 'full' }
13   ];
```

- Update the app-routing.module.ts file to make use of the routes defined above as follows:

```
1    import { NgModule } from '@angular/core';
2    import { CommonModule } from '@angular/common';
3    import { RouterModule, Routes } from '@angular/router';
4
5    import { routes } from './routes';
6
7    @NgModule({
8      imports: [
9        CommonModule,
10       RouterModule.forRoot(routes)
11     ],
12     exports: [ RouterModule ],
13     declarations: []
14   })
15   export class AppRoutingModule { }
```

Using the Routing Module

- Next update the app.component.html file as follows:

```
1    <app-header></app-header>
2    <router-outlet></router-outlet>
3    <app-footer></app-footer>
```

- Then update the app.module.ts file to use the Routing Module as follows:

```
1   . . .
2
3   import { AppRoutingModule } from './app-routing/app-routing.module';
4
5   @NgModule({
6     . . .
7
8       imports: [
9       BrowserModule,
10      FormsModule,
11      HttpModule,
12      MaterialModule,
13      FlexLayoutModule,
14      AppRoutingModule
15    ],
16
17    . . .
18
19  })
20
21  . . .
```

- Finally update the toolbar in the header.component.html file as follows:

```
1   <md-toolbar color="primary">
2     <span><img src="/assets/images/logo.png" height=30 width=41></span>
3     <a md-button routerLink="/home"><span class="fa fa-home fa-lg"></span> Home</a
      >
4     <a md-button><span class="fa fa-info fa-lg"></span> About</a>
5     <a md-button routerLink="/menu"><span class="fa fa-list fa-lg"></span> Menu</a
      >
6     <a md-button><span class="fa fa-address-card fa-lg"></span> Contact</a>
7
8   </md-toolbar>
```

- Save all the changes and do a Git commit with the message "Angular Router Basics".

Conclusions

In this exercise you learnt to set up Angular routing to provide navigation among the views of various components.

Mark as completed