

Angular

Tahaluf Training Center 2021



Chapter 5

- 1 **Shared Module**
- 2 Working on home page
- 3 Input and output



Shared Module

To use module, pipes , services from different modules we will generate **a shared module** contains all modules ,services and pipes which is used from another modules



Shared Module

Example :

```
PS C:\Users\User\Desktop\angular\portalApp> ng g m shared  
CREATE src/app/shared/shared.module.ts (192 bytes)  
PS C:\Users\User\Desktop\angular\portalApp> 
```



Shared Module

In shared module , import all module and declared all component or pipes you will used more than one module.

```
imports: [  
  CommonModule,  
  MatFormFieldModule,  
  FormsModule,  
  ReactiveFormsModule,  
  MatInputModule,  
  MatButtonModule,  
  MatProgressSpinnerModule,  
  NgxSpinnerModule  
],
```



Shared Module

And you must export these module in export array :

```
exports : [  
  MatFormFieldModule,  
  FormsModule,  
  ReactiveFormsModule,  
  MatInputModule,  
  MatButtonModule,  
  MatProgressSpinnerModule,  
  NgxSpinnerModule  
]
```



Chapter 5

- 1 Review
- 2 **Working on home page**
- 3 Input and output



Working on home page

- Before started , we will generate new module for the client called client module .
- The propose of this module is if the client entered to the website successfully (valid email and password in login page) he will go to home page .



Working on home page

- Generate new module called **client**.

```
PS C:\Users\User\Desktop\portalApp> ng g m client --routing
CREATE src/app/client/client-routing.module.ts (249 bytes)
CREATE src/app/client/client.module.ts (280 bytes)
PS C:\Users\User\Desktop\portalApp> |
```



Working on home page

- Inside client module , will generate new component called home component

```
PS C:\Users\User\Desktop\portalApp> ng g c client/home
CREATE src/app/client/home/home.component.html (19 bytes)
CREATE src/app/client/home/home.component.spec.ts (612 bytes)
CREATE src/app/client/home/home.component.ts (267 bytes)
CREATE src/app/client/home/home.component.css (0 bytes)
UPDATE src/app/client/client.module.ts (356 bytes)
PS C:\Users\User\Desktop\portalApp> |
```



Working on home page

Since the auth module is loaded in in app module , client module will loaded in .

In auth –routing.module.ts

```
{  
  path: 'client',  
  loadChildren : ()=> import('../client/client.  
module')  
    .then((m)=>m.ClientModule)  
}
```



Working on home page

- In client –routing.module.ts inside routes array add the default routes.

```
{  
  path: '',  
  component: HomeComponent  
}
```



Working on home page

- If the user logged successfully the button submit will navigate it to the home page , so in **login.component.ts** :

```
submit(){  
  //Go to Loader  
  this.spinner.show();  
  setTimeout(() => {  
    this.spinner.hide();  
    //go to the home page  
    this.router.navigate(['client'])  
  }, 2000)  
}
```



Working on home page

- Now we will add the toolbar from angular material website .
- Add the import module in shared.module .ts :

```
import { MatToolbarModule } from  
'@angular/material/toolbar';
```



Working on home page

Import the **MatToolbarModule** in import section :

In **shared.module.ts** :

```
imports: [  
  CommonModule,  
  ClientRoutingModule,  
  MatToolbarModule  
]
```



Working on home page

In Home.component.html :

```
<mat-toolbar color="primary">  
<mat-toolbar-row>  
<span>Portal App</span>  
</mat-toolbar-row>  
</mat-toolbar>
```



Working on home page

➤ The result will be :



Working on home page

- We will go to add card in home page , before that we will define the **Reusable component** .
- **Reusable component** : the component which is created for one time and use in more than one component



Reusable component

- Generate a reusable component inside home component called PortalCard .

```
PS C:\Users\User\Desktop\angular\portalApp> ng g c client\home\PortalCard
CREATE src/app/client/home/portal-card/portal-card.component.html (26 bytes)
CREATE src/app/client/home/portal-card/portal-card.component.spec.ts (655 bytes)
CREATE src/app/client/home/portal-card/portal-card.component.ts (294 bytes)
CREATE src/app/client/home/portal-card/portal-card.component.css (0 bytes)
UPDATE src/app/client/client.module.ts (694 bytes)
PS C:\Users\User\Desktop\angular\portalApp> |
```



Reusable component

- Inside PortalCard component will creates cards for our PoratlApp example so we will go to angular material and import the pakage of cards in **shared.module.ts** :

```
import { MatCardModule } from  
  '@angular/material/card';
```
- And import MatCardModule in import section .



Reusable component

➤ In PortalCard.component.html :

```
<mat-card class="example-card">
<mat-card-header>
<div mat-card-avatar class="example-header-
image"></div>
<mat-card-title>Front end </mat-card-title>
<mat-card-subtitle>HTML</mat-card-subtitle>
</mat-card-header>

```



Reusable component

```
<mat-card-content>
```

```
<p>
```

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript..

```
</p>
```

```
</mat-card-content>
```

```
  <mat-card-actions>
```

```
    <button mat-button>LIKE</button>
```

```
    <button mat-button>SHARE</button>
```

```
  </mat-card-actions>
```

```
</mat-card>
```



Reusable component

In PortalCode.component.css

```
.example-card {  
    max-width: 400px;  
  
}  
.example-header-image {  
    background-  
    image: url('https://www.onlinecourserereport.com/wp-  
content/uploads/2020/06/shutterstock_153031724-  
1024x768.jpg');  
    background-size: cover;  
  
}
```



Reusable component

- Call PotalApp component in home component and repeat it eight times
Add this code **in home .component .html**

```
<div class=container>  
  <div class="cards">  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
    <app-portal-card></app-portal-card>  
  </div>  
</div>
```



Reusable component

home .component .css

```
.container {  
  width: 98%;  
  margin: 0 auto;  
  margin-top: 40px;  
}  
  
.cards {  
  display: flex;  
  flex-direction: row;  
  justify-content: space-evenly;  
  flex-wrap: wrap;  
}
```



Input and output

- Change the title , sub title and paragraph for each card .
- So, we will sent a permitter from home page component to portalCard components .



Chapter 5

- 1 Review
- 2 Working on home page
- 3 **Input and output**



Input and output

- Change the home page component to :

```
<app-portal-card typeLang="Front End"  
subText="HTML" description="The HyperText Markup  
Language, or HTML is the standard markup language  
for documents designed to be displayed in a web  
browser. It can be assisted by technologies such as  
Cascading Style Sheets (CSS) and scripting languages  
such as JavaScript.." "></app-portal-card>
```



Input and output

- To call typeLang , subTest and description in PortalCard component .
- Define this variable in typescript file In PortalCard.ts inside class :

```
@Input()typeLang:string | undefined;  
@Input()subText:string | undefined;  
@Input()description :string|undefined;  
@Input()photo: string | undefined
```



Input and output

In PortalCard.component.html

```
<mat-card-header>
<div mat-card-avatar class="example-header-
image"></div>
<mat-card-title>{{typeLang}} </mat-card-title>
<mat-card-subtitle>{{subText}}</mat-card-
subtitle></mat-card-header>

<mat-card-content>
<p> {{description}}</p>
</mat-card-content>
```



Input and output

OR you can use array to do this , so in home.component .ts

```
export class HomeComponent implements OnInit {  
  data: any[] = [{  
    typeLang: 'Front End',  
    subText: 'HTML',  
    description: 'The HyperText Markup Language, or  
HTML is the standard markup language for documents  
designed to be displayed in a web browser. It can be  
assisted by technologies such as Cascading Style  
Sheets (CSS) and scripting languages such as  
JavaScript..  
  },
```



Input and output

```
{  
  typeLang: "Back End",  
  subText: "C#",  
  description: " C# (pronounced see  
sharp)[b] is a general-purpose, multi-  
paradigm programming language encompassing  
static typing, strong typing, lexically  
scoped, imperative, declarative, functional,  
generic, object-oriented (class-based), and  
component-oriented programming  
disciplines.[15]"  
}  
  
]
```



Input and output



شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

In home.component .html

```
<div class=container>
<div class="cards">
<app-portal-card
[typeLang]="data[0].typeLang"
[subText]="data[0].subText"
description="data[0].description"></app-
portal-card>
<app-portal-card
[typeLang]="data[1].typeLang"
[subText]="data[1].subText"
description="data[1].description"></app-
portal-card>
```



Input and output

- Update home component and use less code using for loop in html
- Syntax :

<*ngFor="let variable of array-name">



Input and output

In home.component.html

```
<div class =container>
<div class ="cards">
<app-portal-card *ngFor ="let card of data"
[typeLang]="card.typeLang" [subText]="card
.subText" description="card.description">
</app-portal-card>
</div>
</div>
```



Input and output

Write Event

If the user click to the Language will navigate to the profile for this language .



Input and output

Exercise

Generate new component in client module called profile and give it a route



Input and output

Solution

```
PS C:\Users\User\Desktop\angular\portalApp> ng g c client\profile
CREATE src/app/client/profile/profile.component.html (22 bytes)
CREATE src/app/client/profile/profile.component.spec.ts (633 bytes)
CREATE src/app/client/profile/profile.component.ts (279 bytes)
CREATE src/app/client/profile/profile.component.css (0 bytes)
UPDATE src/app/client/client.module.ts (780 bytes)
PS C:\Users\User\Desktop\angular\portalApp> |
```



Input and output

Solution

In client-routting.module.ts

```
const routes: Routes = [  
  {  
    path: '',  
    component: HomeComponent  
  },  
  {  
    path: 'profile',  
    component: ProfileComponent  
  }  
];
```



Input and output

In home .component.ts

```
goTopprofile(){
```

```
this.router.navigate(['profile'])  
}
```



Input and output

Update home component

```
<div class=container>
  <div class="cards">
    <app-portal-card *ngFor="let card of data"
      [typeLang]="card.typeLang"
      [subText]="card.subText"
      description="card.description"
      (openProfile)=" goToprofile()"></app-
portal-card>

  </div>
</div>
```



Input and output

In portalCards.component.ts add in the class

```
@Output()openProfile=new EventEmitter();
```

And add this function to call
openprofile method

```
showCoursePorfile(){  
    //call openProfile method();  
    this.openProfile.emit();  
}
```

