Angular

Tahaluf Training Center 2021

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

Chapter 6

1 **Services**

2 **Pipeline**

Services

Angular services help is intended to exemplify business rationale and information with various components of Angular.

Services

In order to create services:

In terminal:

ng generate services Folder_name/Service_name

or

ng g s folder_name/service_name

Services

Generate new services called home.

```
PS C:\Users\User\Desktop\Training\TrainingWebSite> ng g s services/home
```

In the home service define a string and read it in home component.

In home.service.ts

```
export class HomeService {
    message: string =
"This is from home service"
    constructor() {
    }
}
```

Services

Read this services from home component.
First: define an object of services in home.component.ts as a parameter of the constructor.

In home.component.ts

```
constructor(private router: Router,
public homeServices : HomeService) { }
```

Then in home.component.ts

```
<div class=container>
    <h2>Welcome to our Portal</h2>
    <h2>{{ homeServices.message }}</h2>
</div>
```

Exercise :

Read the home service from login component and if the user logged successfully, update the message to “You are logged In ”.

Exercise Solution:

In login.component.ts define an object of the services:

```
constructor(private spinner:
NgxSpinnerService, private router: Router,
    public homeservices: HomeService)
{
}
```

In login.component.html:

```
<h2>{{homeservices.message}}</h2>
```

Exercise Solution:

In login.component.ts, in submit function:

```
submit(){
        //Go to Loader
        this.spinner.show();
        setTimeout(() => {
            this.spinner.hide();
            this.homeservices.message =
    "You are logged In"
        //go to the home page

    this.router.navigate(['client'])
        }, 2000)
    }
```

Now , we will define an array in home services called selectedCourse , and if the user enter for the type of this course will navigate to profile page and load the data for this course.

In the homeServices:

```
selctorCours :any = {};
```

In Course-card.ts :

```
showCoursePorfile(){
    this.homeservice.selctorCours = {
        typeLang: this.typeLang,
        description: this.description,
        subtitle: this.subText
    }

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

In profile.component.ts: define the home services.

```
constructor(public homeServices:
HomeService) { }
```

In profile.component.html:

```
<h2>Welcome to {{homeServices.selctorCours.
subtitle}}profile</h2>
<h2>Type of course is {{homeServices.selcto
rCours.
typeLang}}</h2>
<h2>bio: {{homeServices.selctorCours.descri
ption}}
</h2>
```

The goal of creating the service is to reduce the writing of code and arrange it so that we reach the best practices.

So all the logic must be written inside the service.

The logic in home component is the array
so , remove the array from home component and rewrite it in home services.

Update the home.component.html:

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

Services

The logic of the login component, create a new services called auth.

Remove the submit body from login.ts and rewrite it in auth services.

Note: You can defined services or package inside another services like NgSpinnerServices and Route package.

In auth services :

```
constructor(private spinner: NgxSpinnerService,
    private router: Router,
    private homeservices: HomeService) { }
login(email: any, password: any){
    console.log(email, password)
    //Go to Loader
    this.spinner.show();
    setTimeout(() => {
        this.spinner.hide();
        this.homeservices.message = "You are logged In"
        //go to the home page
        this.router.navigate(['client'])    }, 2000)
}
}
```

In login.component.ts:

```
submit(){

this.authServices.login(this.emailFormContr
ol, this.passwordFormControl)
}
```

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

Chapter 6

1 Services

2 Pipeline

Pipes are a feature in Angular.
They are a simple way to transform values in an Angular template (In html tags).

There are some built in pipes like date using | date, uppercase using |uppercase and lowercase using
| lowercase.

Built in Pipes in angular:

1. **date** which return formatted date.
2. **uppercase** which return upper case formatted.
3. **lowercase** which return lowercase formatted.
4. **percent** which convert a value to a percentage.

Example for using date pipes:

In homeServices.ts :

```
{
    typeLang: 'HTML',
    subText: new Date(),},
```

In Course-card.component.html :

```
<mat-card-subtitle>{{subText | date}}</mat-card-subtitle>
```

Example for using uppercase pipes:

In homeServices.ts:

```
{
        typeLang: 'HTML',
        subText:’ html',},
```

In Portal-card.component.html :

```
<mat-card-subtitle>{{subText |
uppercase}}</mat-card-subtitle>
```

Example for using percent pipes:

In homeServices.ts:

```
{
    typeLang: 'HTML',
    subText: 55,},
```

In Portal-card.component.html :

```
<mat-card-subtitle>{{subText | percent}}</mat-
card-subtitle>
```

Since there are built in pipes , you can also make a custom pipe.

The syntax to generate new pipes is :

```
ng g p folder_name/pipe_name
```

Generate a new pipeline called dateFormate inside a Pipes folder.

```
PS C:\Users\User\Desktop\Training\TrainingWebSite> ng g p pipes/dateFormate
CREATE src/app/pipes/date-formate.pipe.spec.ts (208 bytes)
CREATE src/app/pipes/date-formate.pipe.ts (227 bytes)
UPDATE src/app/app.module.ts (475 bytes)
```

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

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

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

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

And you must export these module in export array :

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

Inside dateFormat pipe

```
transform(value: string, ...args: unknown[]):
unknown {
    const date=new Date(value);
    //day/month/year
const formattedDate=`${date.getDate()}/$
{date.getMonth()+1}/$ {date .getFullYear}`;
return formattedDate;
}
```

Pipeline

In home services

```
{
    typeLang: 'HTML',
    subText: new Date(),},
```

And in portalCard.component.html:

```
<mat-card-subtitle > {

{
    subText |dateFormat
}

</mat-card-subtitle >
```