

Angular

Tahaluf Training Center 2021



Chapter 6

1 Services

2 Pipeline



Services

Services

Angular services 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\angular\portalApp> ng g s services/home  
CREATE src/app/services/home.service.spec.ts (347 bytes)  
CREATE src/app/services/home.service.ts (133 bytes)  
PS C:\Users\User\Desktop\angular\portalApp> |
```



Services

Example

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

- To 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) { }
```



Services

Then in home .component.html

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



Services

Exercise :

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



Services

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>
```



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)  
}
```



Services

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.ts` :

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



Services

In Portal-card.ts :

```
showCoursePorfile(){  
    this.homeservice.selctorCours = {  
        typeLang: this.typeLang,  
        description: this.description,  
        subtitle: this.subText  
    }  
  
    //call openProfile method();  
    this.openProfile.emit();  
}  
}
```



Services

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.selctorCours.
typeLang}}</h2>
<h2>bio: {{homeServices.selctorCours.description}}
</h2>
```



Services

- 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.



Services

- 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-portal-card *ngFor="let card of  
homeServices.data" [typeLang]="card.typeLang"  
[subText]="card.subText"  
description="card.description" (openProfile)="  
goTopprofile()"></app-portal-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 .
- **Notes** : You can defined services or package inside another services like NgSpinnerServices and Route package.



Services

In auth services.ts :

```
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)  
}  
}
```



Services

In login.component.ts:

```
submit(){  
  
  this.authService.login(this.emailFormControl,  
    this.passwordFormControl)  
}
```



Chapter 6

1 Services

2 Pipeline



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**.



Pipeline

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



Pipeline

➤ **Example** for using date pipes:

In homeServices.ts :

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

In Portal-card.component.html :

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



Pipeline

➤ **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>
```



Pipeline

➤ **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>
```



Pipeline

- 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



Pipeline

- Generate a new pipeline called dateFormat inside a Pipes folder

```
PS C:\Users\User\Desktop\angular\portalApp> ng g p Pipes/dateFormat
CREATE src/app/Pipes/date-format.pipe.spec.ts (204 bytes)
CREATE src/app/Pipes/date-format.pipe.ts (225 bytes)
UPDATE src/app/app.module.ts (832 bytes)
PS C:\Users\User\Desktop\angular\portalApp> |
```



Pipeline

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



Pipeline

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 >
```

