Harmony IT Solution

Angular

Tahaluf Training Center 2022

1 Services in Angular.

2 Pipes.

3 RXJS Library.

Objective

The Objective of this lecture

- Understand the concept of the service and how to generate a new service in the terminal.
- How to use the pipe in the template in the angular project.
- Understand the RXJS and the concept of observable methods.

Services in Angular

Overview of Services

Services are reusable pieces of code with a defined purpose. Your application will use this code across many components.

Overview of Services

Components need to access data. The data access code can be written in each component, but this is inefficient and violates the principle of single responsibility.

Rather, the component should focus on how to present data to the user.

Overview of Services

The process of getting data from the back-end server is delegated to another class, called the Service class.

This class provides data to every component that requires it.

The purpose of Angular Services

- Is to provide features independent of components such as logging services.
- Sharing logic or data among components.
- Encapsulate interactions with external parties, such as data access.

Advantages of Angular Service

○ Tests are easier on services.

○ Debugging is easier.

○ The service can be reused in many places.

How to generate a new service

To create a service, use this command on the terminal:

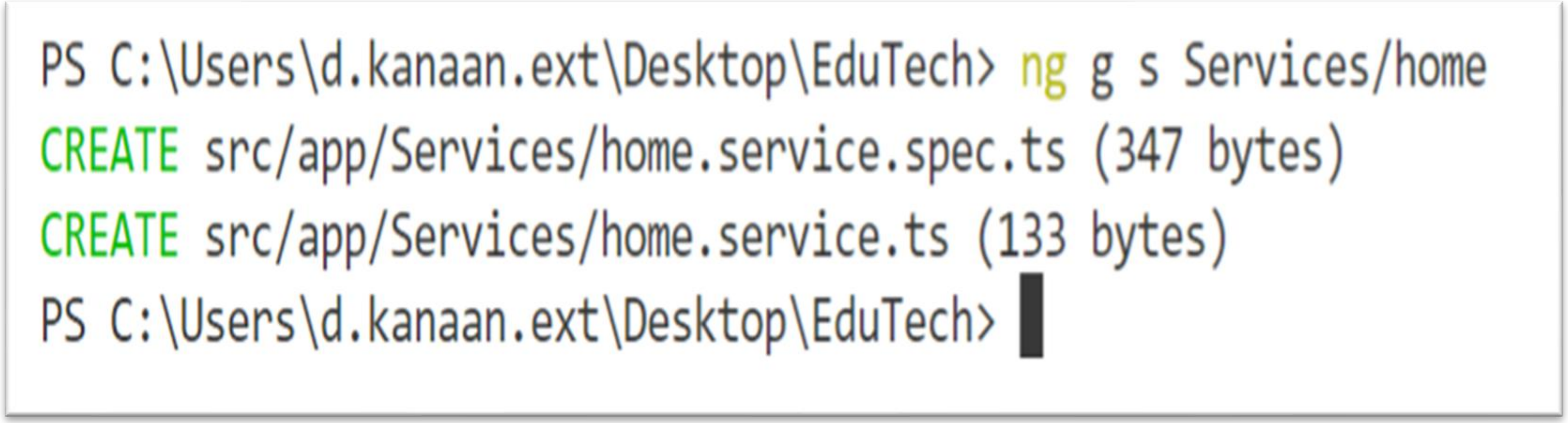
```
ng generate service service_name
```

or

```
ng g s service_name
```

For example, to generate a new service

Generate a new service called home.
Here, the Services is a folder that will contain a home service.

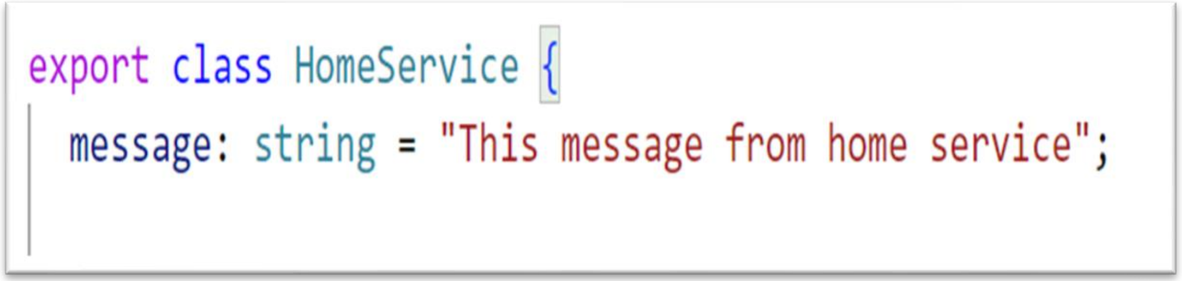


```
PS C:\Users\d.kanaan.ext\Desktop\EduTech> ng g s Services/home
CREATE src/app/Services/home.service.spec.ts (347 bytes)
CREATE src/app/Services/home.service.ts (133 bytes)
PS C:\Users\d.kanaan.ext\Desktop\EduTech> █
```

For example, to generate a new service

In the home service, define a new property and read the value of it in the home component.

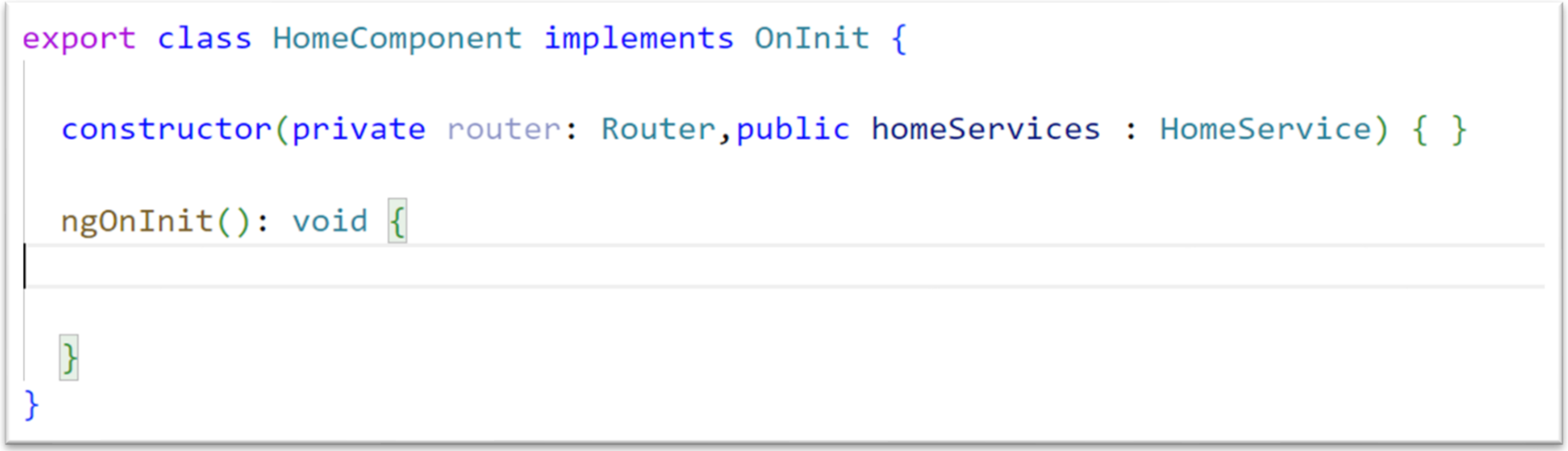
In home.service.ts:



```
export class HomeService {
  message: string = "This message from home service";

```

For example, to generate a new service

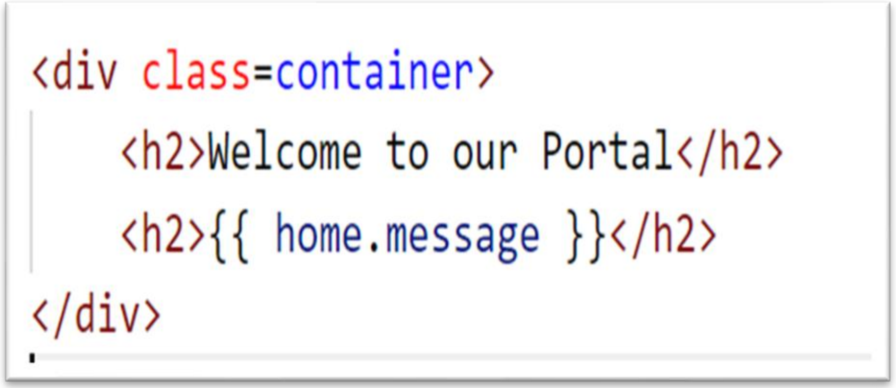
Define an object of home service in home.component.ts as a parameter of the constructor.



```
export class HomeComponent implements OnInit {

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

  ngOnInit(): void {


  }
}
```

For example, to generate a new service

Read the value in home.component.ts.



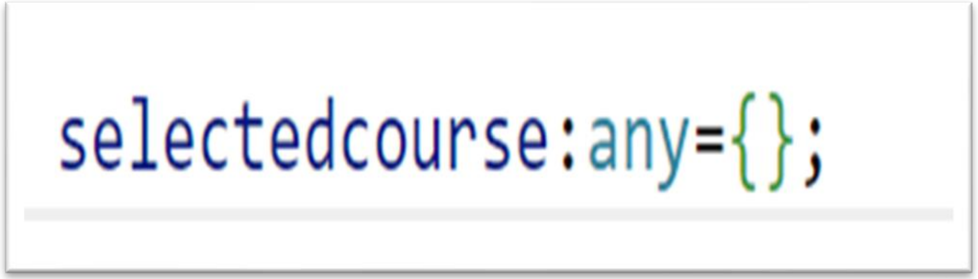
```
<div class=container>
    <h2>Welcome to our Portal</h2>
    <h2>{{ home.message }}</h2>
</div>
```

Exercise:

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

Services Example

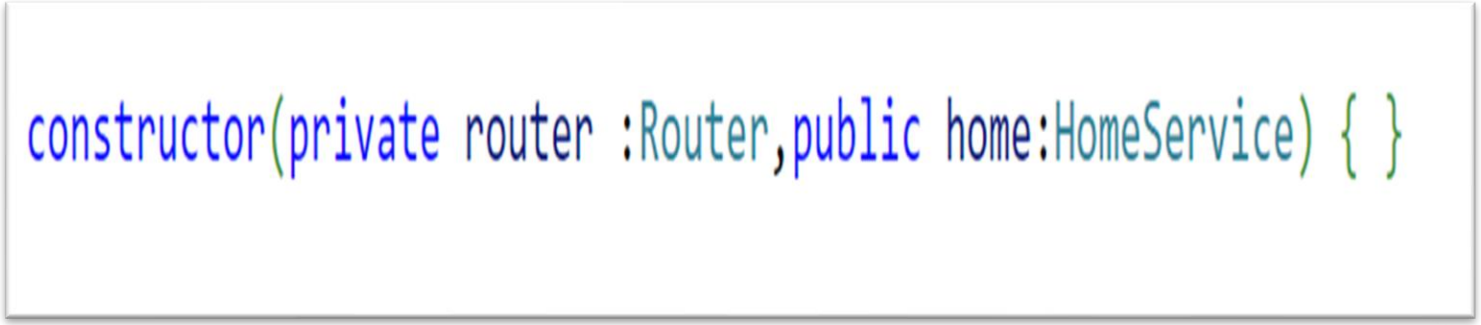
Define an object in-home services called selected course, and if the user clicks on the image of the course, will navigate to the profile page and load the data for this course.



```
selectedcourse:any={};
```

Services Example

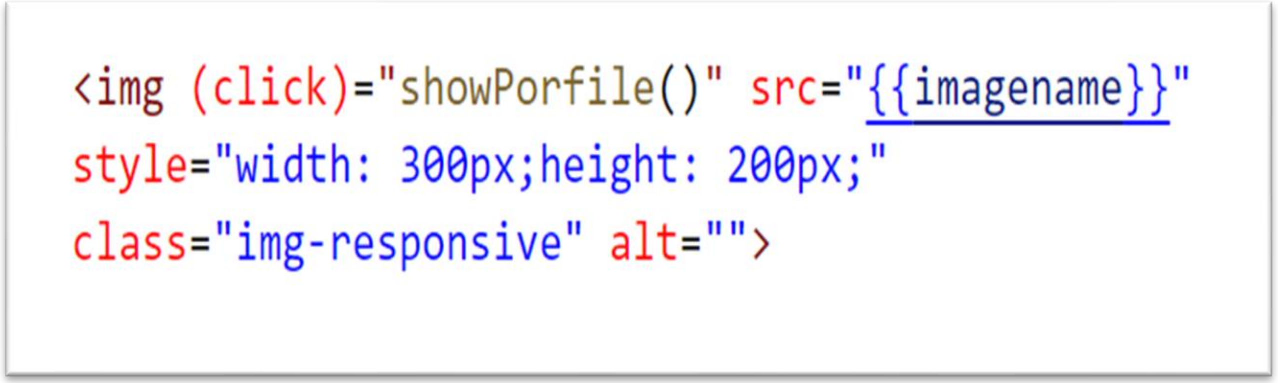
Define an object of home service and router class service.



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

Services Example

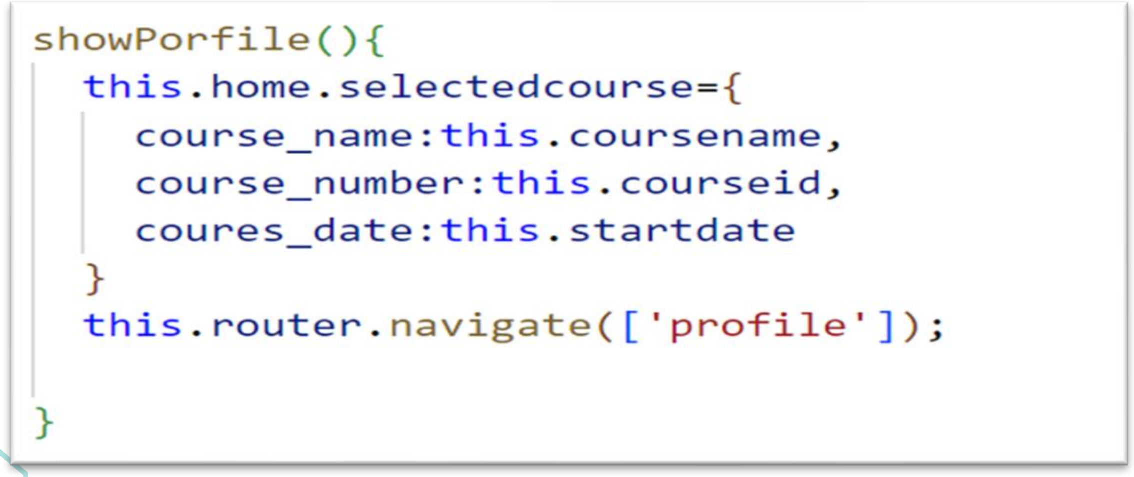
In course-card.html add the output event of the image element.



```
<img (click)="showPorfile()" src="{{imagename}}"
style="width: 300px;height: 200px;"
class="img-responsive" alt="">
```

Services Example

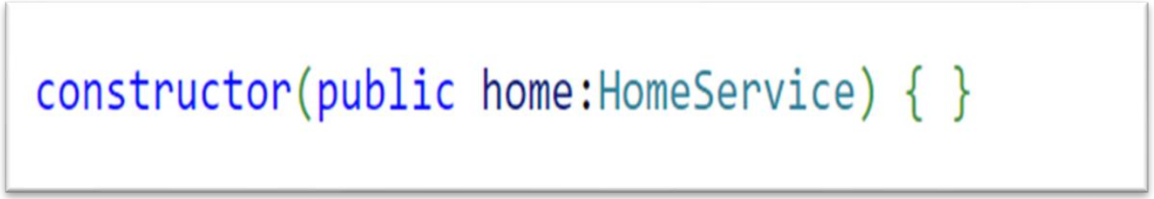
Implement the show profile function in the typescript file for the course card component.
In course-card.ts



```
showPorfile(){
  this.home.selectedcourse={
    course_name:this.coursename,
    course_number:this.courseid,
    coures_date:this.startdate
  }
  this.router.navigate(['profile']);

}
```

Services Example

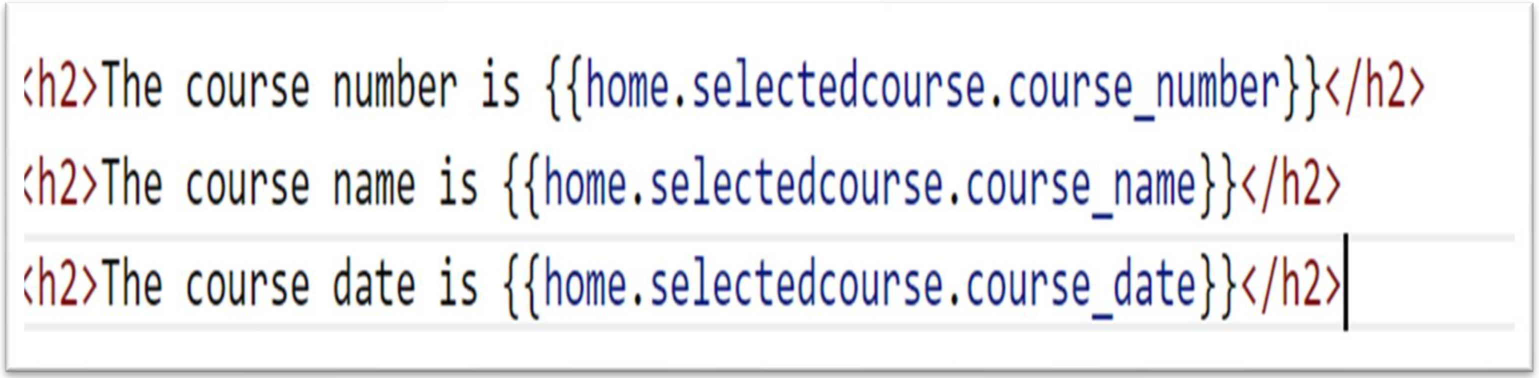
Define an object of home service in profile.component.ts and use this object to bind the data in the profile component



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

Services Example

In profile. component.html

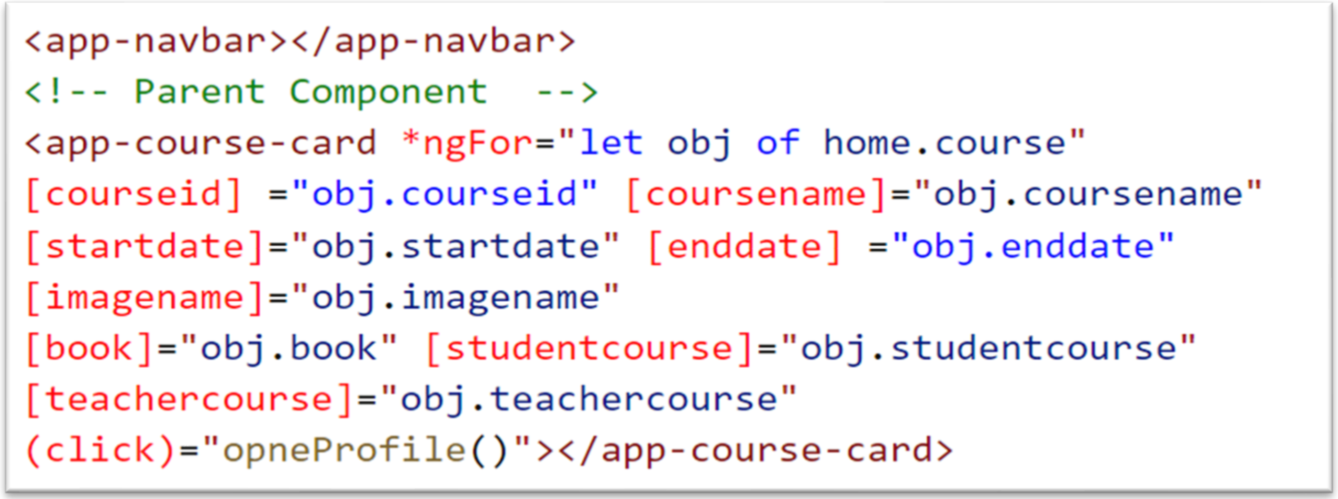


```
<h2>The course number is {{home.selectedcourse.course_number}}</h2>
<h2>The course name is {{home.selectedcourse.course_name}}</h2>
<h2>The course date is {{home.selectedcourse.course_date}}</h2>
```

Services Example

The logic in the home component is the array. So, remove the array from the home component and rewrite it in-home service.

Update the home.component.html:



```
<app-navbar></app-navbar>
<!-- Parent Component  -->
<app-course-card *ngFor="let obj of home.course"
[courseid] ="obj.courseid" [coursename]="obj.coursename"
[startdate]="obj.startdate" [enddate] ="obj.enddate"
[imagename]="obj.imagename"
[book]="obj.book" [studentcourse]="obj.studentcourse"
[teachercourse]="obj.teachercourse"
(click)="opneProfile()"></app-course-card>
```

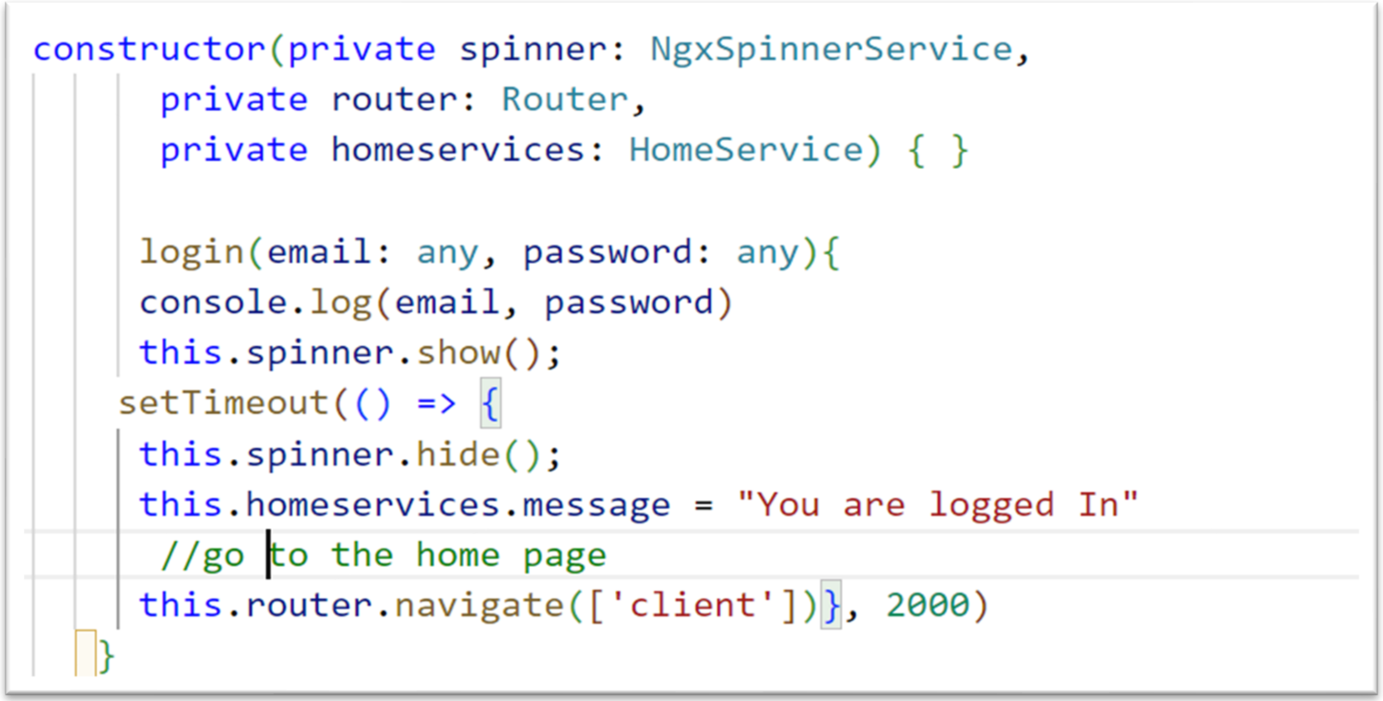
Services Example

Create a new service called auth. And remove the body of the submit function from login.ts and rewrite it in auth service.

NOTE: You can define services or package inside another service like NgSpinnerServices and Route package.

Services Example

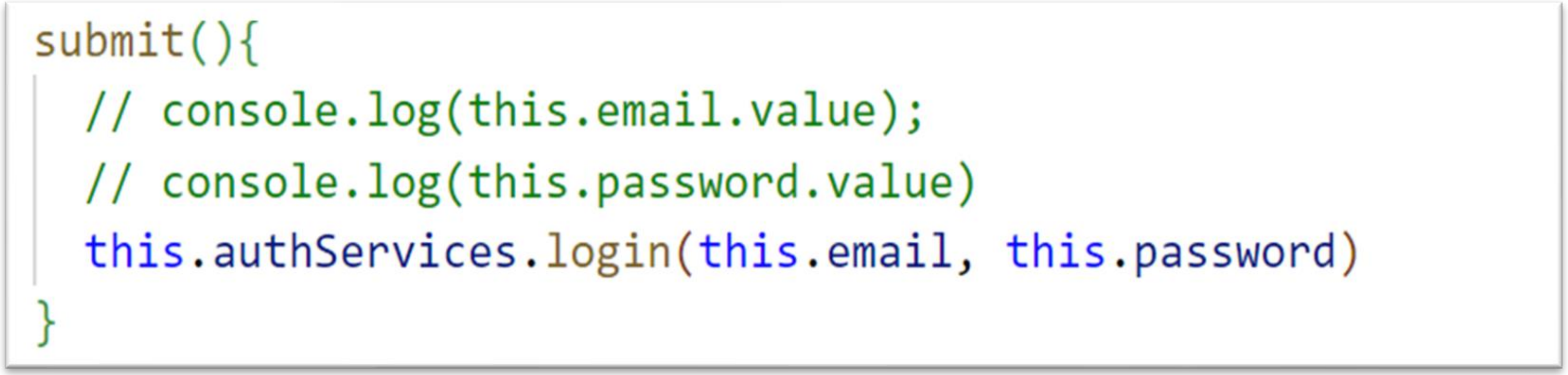
In auth.service.ts



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

     login(email: any, password: any){
     console.log(email, password)
     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 Example

Update the login component, so in login. component.ts



```
submit(){
  // console.log(this.email.value);
  // console.log(this.password.value)
  this.authServices.login(this.email, this.password)
}
```

Pipes

Overview of Pipes

Angular has built-in pipes:

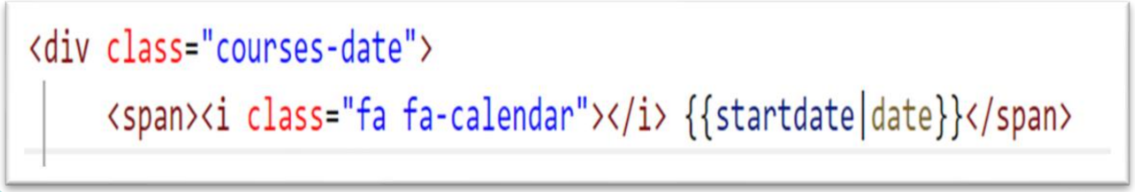
- Date: returns a formatted date.
- Uppercase: return upper case formatted.
- Lowercase: returns lowercase formatted.
- Percent: converts a value to a percentage.

Pipes Example

In homeService.ts:

```
{
  course_name: 'HTML',
  startdate: new Date(),
},
```

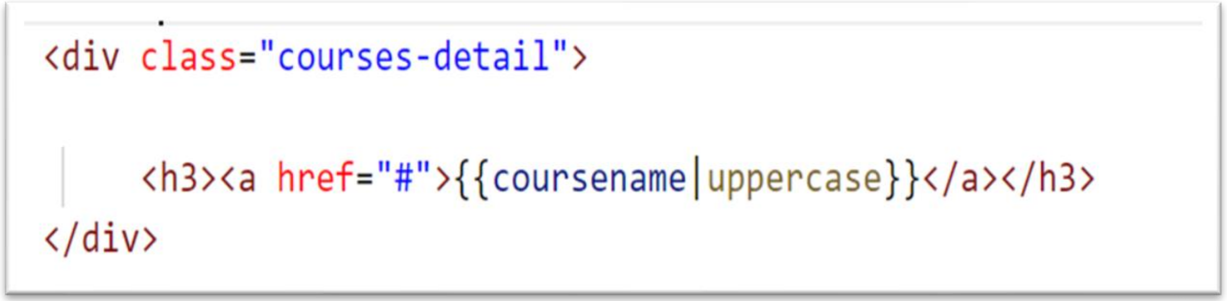
In course-card.component.html:



```
<div class="courses-date">
    <span><i class="fa fa-calendar"></i> {{startdate|date}}</span>
```

Pipes Example

In course-card.component.html:



```
<div class="courses-detail">

    <h3><a href="#">{{coursename|uppercase}}</a></h3>
</div>
```

RxJS

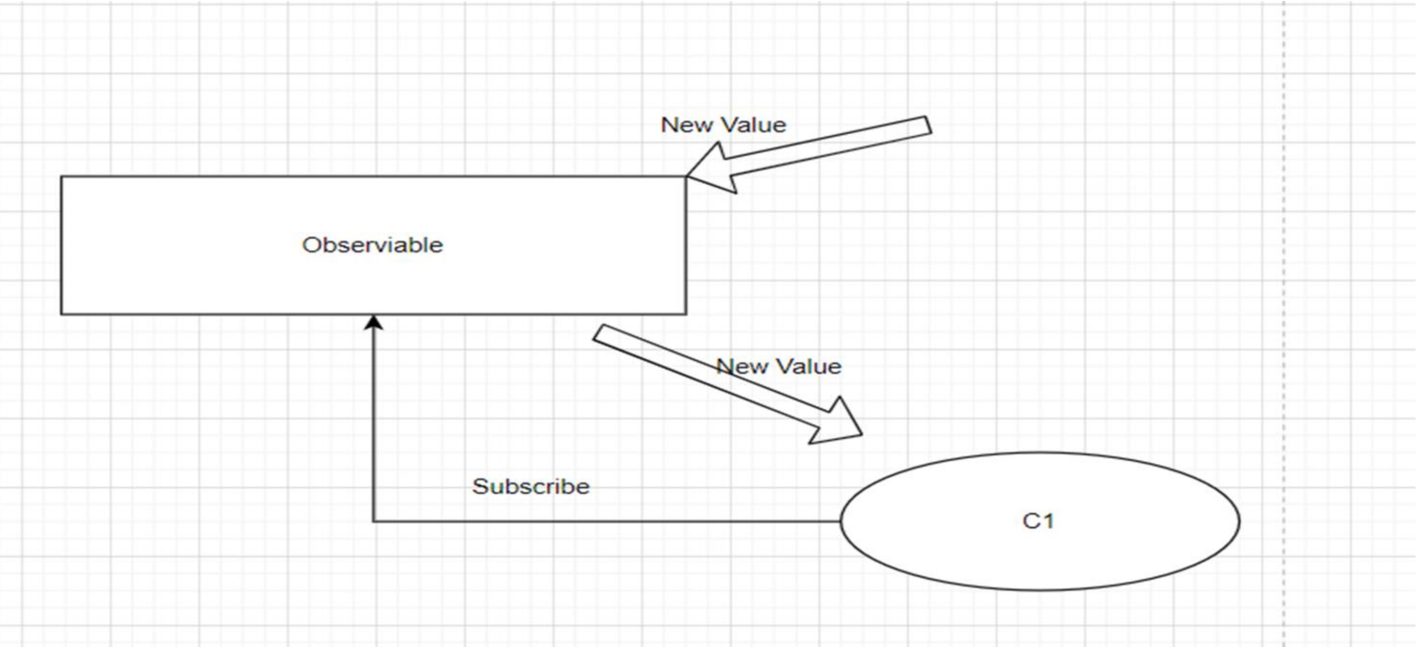
Overview of RxJS:

Reactive extensions for JavaScript are called RxJS.

This JavaScript library uses observables to handle asynchronous calls, callbacks, and event-based programming using reactive programming.

Overview of Observable:

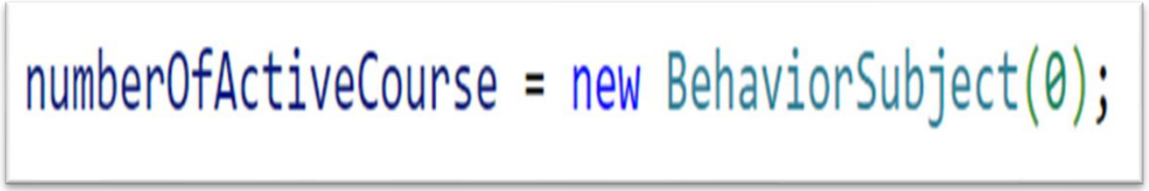
-



RxJS Example:

In HomeServices, declare an object of behavior subject and give the default value = 0.

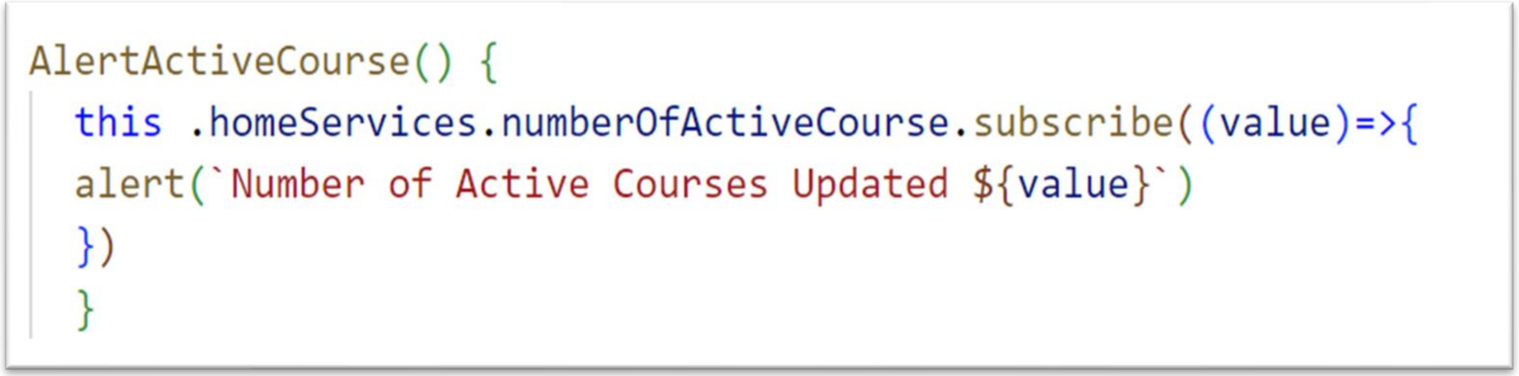
This object is like a counter for how many courses are in the system.



```
numberOfActiveCourse = new BehaviorSubject(0);
```

RxJS Example:

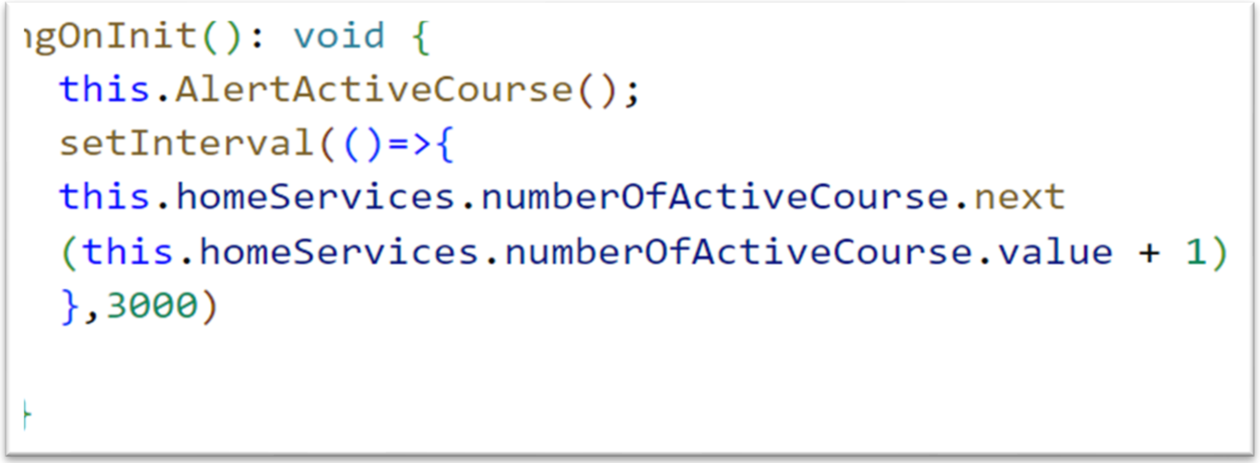
Create a function in home.component.ts to subscribe to the object of the behavior subject.



```
AlertActiveCourse() {
  this .homeServices.numberOfActiveCourse.subscribe((value)=>{
  alert(`Number of Active Courses Updated ${value}`)
  })
  }
```

RxJS Example:

Inside OnInit function in **home.component.ts**:



```
ngOnInit(): void {
  this.AlertActiveCourse();
  setInterval(()=>{
  this.homeServices.numberOfActiveCourse.next
  (this.homeServices.numberOfActiveCourse.value + 1)
  },3000)


}
```

Exercise:

Add another variable in home.component.ts and make the value for this variable equal to the value for the number of active courses and display it on the home component and In profile component.

References

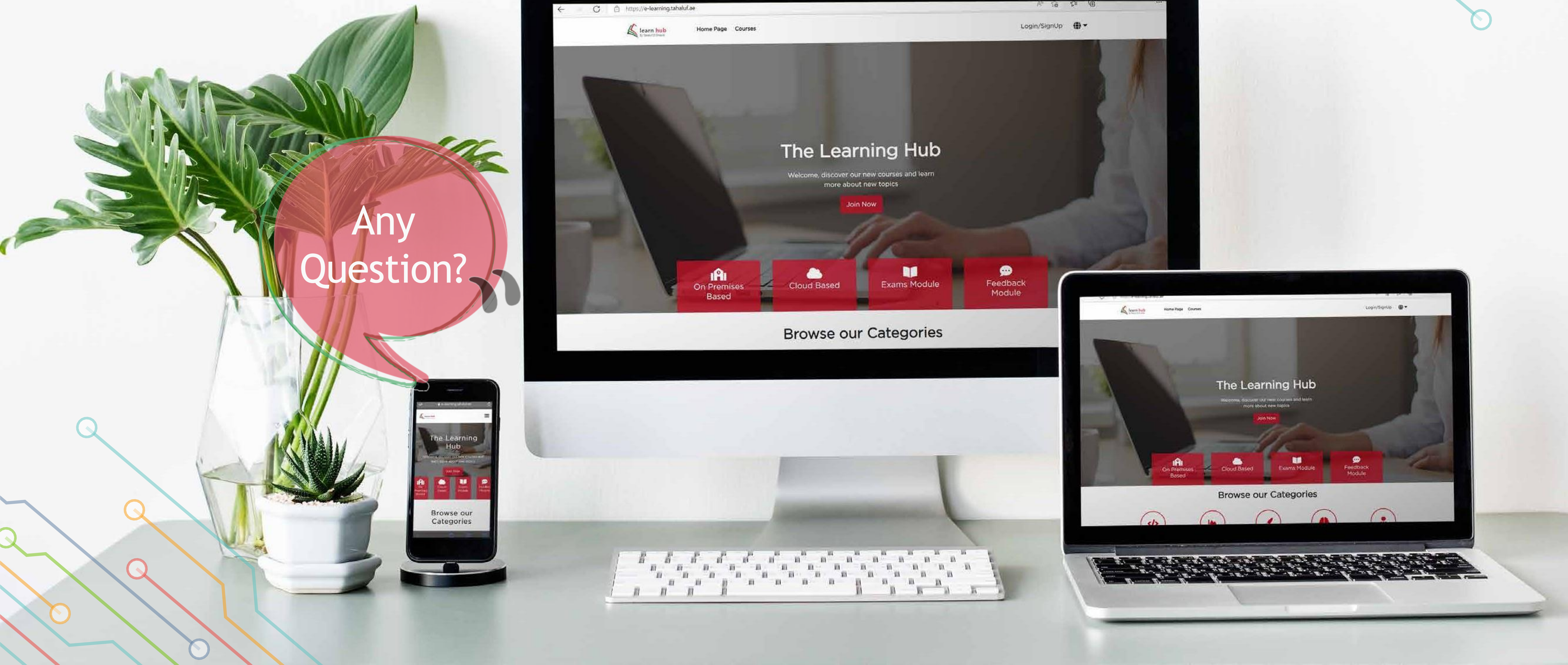
[1] Angular, “Angular,” *Angular.io*, 2019. https://angular.io/

[2] “Complete Angular Tutorial For Beginners,” *TekTutorialsHub*. https://www.tektutorialshub.com/angular-tutorial/

[3]“npm | build amazing things,” *Npmjs.com*, 2019. https://www.npmjs.com/

[4]“Angular Tutorial for Beginners | Simplilearn,” *Simplilearn.com*. https://www.simplilearn.com/tutorials/angular-tutorial (accessed Aug. 19, 2022).

Harmony IT Solution



Any Question?