

## Angular Services

Services in Angular are responsible for handling business logic, HTTP calls, data storage, and any other reusable function that does not involve displaying content on the screen (UI).

### Examples:

A service that logs in and out

A service that searches for products in the API

A service that saves data to localStorage

### Features

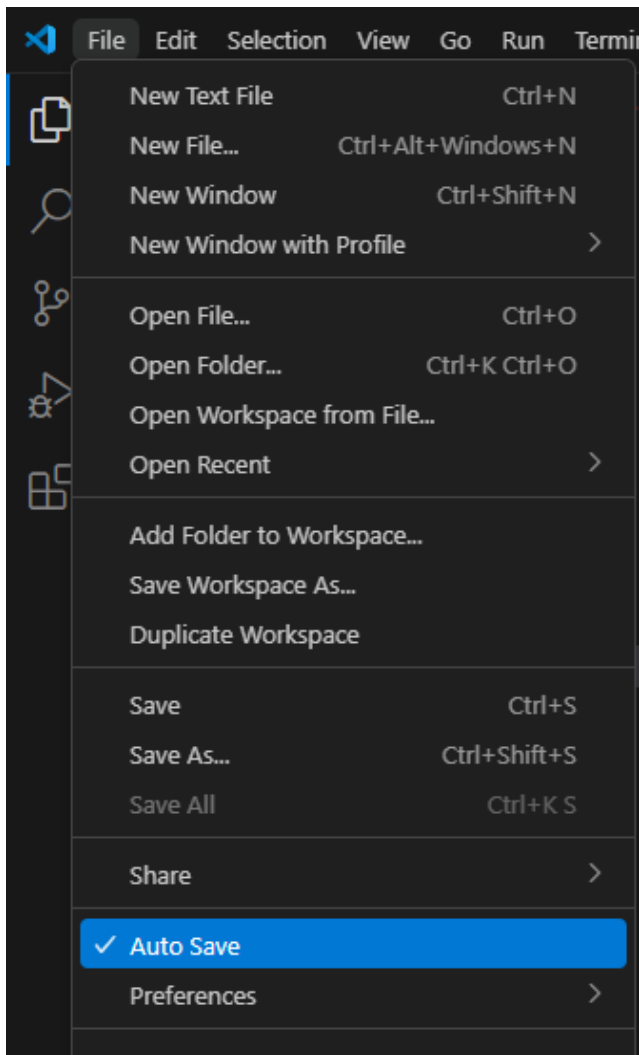
- A service is a class
- Decorated with Injectible

@Injectable lets Angular know how to create and provide an instance of this class when needed.

- They share the same piece of code
- Hold the business logic
- Interact with the backend
- Share data among components
- Services are singleton
- Registered on modules or components

## ❤️ VSCode tip

Configure your Visual Code to auto save in File > Auto Save



## ❤️ Development

Employee detail list

*ng new hello-world*

**\*\*Creating a component\*\***

*ng g c e-info*

Verify that folder assets is on the angular.json to add an image

```
"assets": [  
  "src/assets",  
  "src/favicon.ico"  
],
```

Import the component in app.component.ts

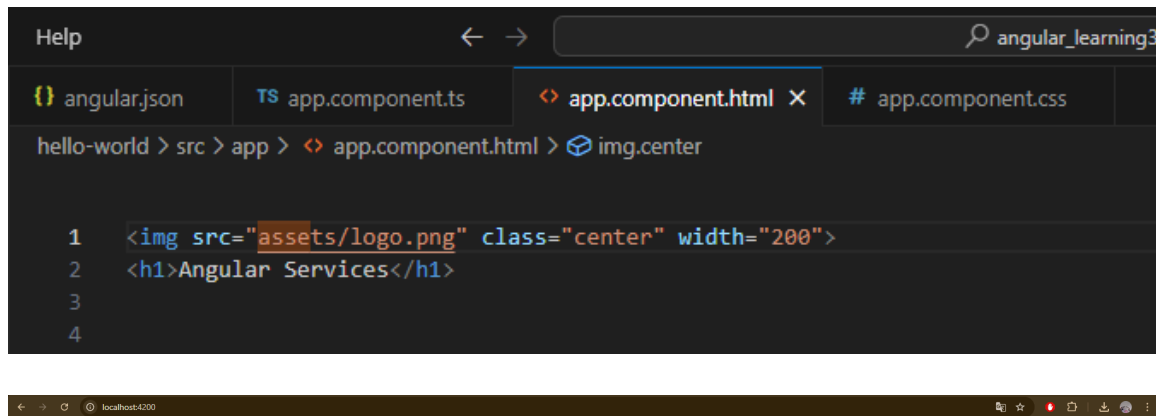
```
import { Component } from '@angular/core';  
  
import { RouterOutlet } from '@angular/router';  
  
import { EInfoComponent } from './e-info/e-info.component';  
  
@Component({  
  selector: 'app-root',  
  imports: [RouterOutlet, EInfoComponent],  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
  
export class AppComponent {  
  title = 'hello-world';  
}
```

**\*\*Create a service\*\***

*ng g service data*

```
app.component.ts  
TS data.service.spec.ts  
TS data.service.ts
```

Initial app.component.html



Angular Services

**\*\*On Init setting\*\***

OnInit in Angular is a lifecycle hook. It allows you to run some code as soon as the component is initialized.

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-e-info',
  imports: [],
  templateUrl: './e-info.component.html',
  styleUrls: ['./e-info.component.css']
})
export class EInfoComponent implements OnInit {
```

```

infoReceived1 : string[] = [];

infoReceived2 : string[] = [];

infoReceived3 : string[] = [];


//methods

getInfoReceived1() {}

getInfoReceived2() {}

getInfoReceived3() {}

constructor() {}

ngOnInit(): void {

}

}

```

**\*\*Creating service records\*\***

```

import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})

export class DataService {

  info1:string[] = ["John", 'E354', 'j@abc.net']

  info2:string[] = ["Rob", 'E673', 'rb@abc.net']

  info3:string[] = ["Rose", 'E865', 're@abc.net']


//methods that return this employees infos

```

```

getinfo1():string[]{
    return this.info1
}

getinfo2():string[]{
    return this.info2
}

getinfo3():string[]{
    return this.info3
}

constructor() { }
}

```

**\*\*Dependency injection\*\***

Import and add in providers

```

1  import { Component, OnInit } from '@angular/core';
2  import { DataService } from '../data.service';
3
4  @Component({
5      selector: 'app-e-info',
6      providers: [DataService],
7      templateUrl: './e-info.component.html',
8      styleUrls: ['./e-info.component.css']
9  })
10 export class EInfoComponent implements OnInit {
11

```

```

import { Component, OnInit } from '@angular/core';

import { DataService } from '../data.service';

```

```
@Component({
  selector: 'app-e-info',
  providers: [DataService],
  templateUrl: './e-info.component.html',
  styleUrls: ['./e-info.component.css']
})

export class EInfoComponent implements OnInit {

  infoReceived1 : string[] = [];
  infoReceived2 : string[] = [];
  infoReceived3 : string[] = [];

  //methods

  getinfoReceived1(){
    this.infoReceived1 = this.dservice.getinfo1();
  }

  getinfoReceived2(){
    this.infoReceived2 = this.dservice.getinfo2();
  }

  getinfoReceived3(){
    this.infoReceived3 = this.dservice.getinfo3();
  }

  constructor( private dservice:DataService){}

  ngOnInit(): void {
  }
}
```

```
}
```

**\*\*Creating the UI\*\***

e-info.component.html

```
<button type="button" name="button"
(click)="getinfoReceived1()">Employee 1</button>

<ul class="list-group">

  <li *ngFor = "let info of infoReceived1"
class="list-group-info">{{info}}</li>

</ul>

<button type="button" name="button"
(click)="getinfoReceived2()">Employee 2</button>

<ul class="list-group">

  <li *ngFor = "let info of infoReceived2"
class="list-group-info">{{info}}</li>

</ul>

<button type="button" name="button"
(click)="getinfoReceived3()">Employee 3</button>

<ul class="list-group">

  <li *ngFor = "let info of infoReceived3"
class="list-group-info">{{info}}</li>

</ul>
```

Include the selector to show the component at app.component.html

```

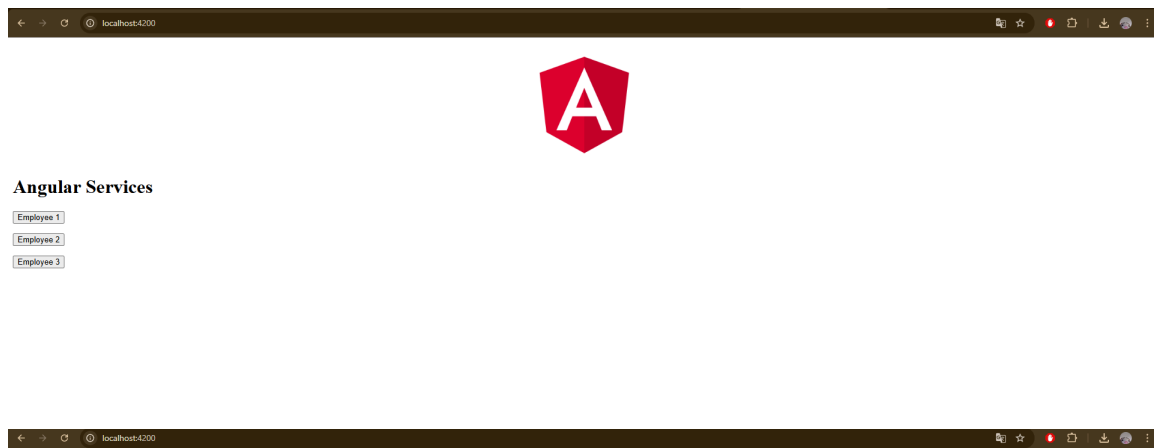
```



```
<h1>Angular Services</h1>
```

```
<app-e-info></app-e-info>
```

**\*\*Import the CommonModule to use \*ngFor**



**\*\*Add another info in a form to the employees\*\***

Service can be used to update, modify, delete and more.

This #frm="ngForm" is creating a local reference (var) to the form in your template and associating that reference with the Angular form.

```
<form #frm = "ngForm">

  <div class="">

    <input type="text" name="location" value="" ngModel>

    <button type="submit" name="button">Add info</button>

  </div>

</form>
```

```
<button type="button" name="button"
(click)="getinfoReceived1()">Employee 1</button>

<ul class="list-group">

  <li *ngFor = "let info of infoReceived1"
class="list-group-info">{{info}}</li>

</ul>
```

```
<button type="button" name="button"
(click)="getinfoReceived2()">Employee 2</button>

<ul class="list-group">

  <li *ngFor = "let info of infoReceived2"
class="list-group-info">{{info}}</li>

</ul>
```

```
<button type="button" name="button"
(click)="getinfoReceived3()">Employee 3</button>

<ul class="list-group">

  <li *ngFor = "let info of infoReceived3"
class="list-group-info">{{info}}</li>

</ul>
```

**\*\*Importing Forms module\*\***

e-info.component.ts

```
import { Component, OnInit } from '@angular/core';

import { DataService } from '../data.service';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

@Component({

  selector: 'app-e-info',

  imports: [CommonModule, FormsModule],

  providers: [DataService],

  templateUrl: './e-info.component.html',

  styleUrls: ['./e-info.component.css']

})
```

**\*\*Creating another method to include this form information to the employees\*\***

e-info.component.ts

```
36
37   updateInfo(frm :any){
38     this.dservice.addInfo(frm.value.location)
39   }
40
```

Add the addInfo in the service

```
addInfo(info: string) {

  this.info1.push(info);

  this.info2.push(info);

  this.info3.push(info);

}
```

```
    return this.info1  
  }  
}
```

Include the method in the form

```
angular.json  e-info.component.html x  TS app.component.ts  app.component.html  TS data.service.spec.ts  TS e  
hello-world > src > app > e-info > e-info.component.html > ul.list-group  
1  <form #frm = "ngForm" (ngSubmit)="updateInfo(frm)">  
2    <div class="">  
3      <input type="text" name="location" value="" ngModel>  
4      <button type="submit" name="button">Add info</button>  
5    </div>  
6  </form>  
7
```



## Angular Services

- John
- E354
- j@abc.net

- Rob
- E673
- rb@abc.net

- Rose
- E865
- re@abc.net

# Angular Services

test	Add info
Employee 1	

- John
- E354
- j@abc.net
- test

Employee 2
------------

- Rob
- E673
- rb@abc.net
- test

Employee 3
------------

- Rose
- E865
- re@abc.net
- test

## Angular and Bootstrap

Bootstrap is a free, open-source front-end framework used to build responsive, mobile-first websites. It provides a collection of pre-designed HTML, CSS, and JavaScript templates for typography, forms, buttons, tables, navigation, and other interface components. This framework simplifies web development by offering a robust set of components and a responsive grid system, allowing developers to create visually appealing and consistent web interfaces quickly.

## ♥ Embedding Bootstrap into Angular

- Bootstrap CDN - is a public content delivery network. It enables users to load CSS, JavaScript and images remotely from its services.
- Bootstrap NPM - Bootstrap, JQuery and other packages and frameworks can be installed on the command prompt.

## ♥ Development

**\*\*Creating a new angular project\*\***

*ng new democomponent*

**\*\*install jquery and bootstrap\*\***

```
C:\Users\Gabriel\Desktop\angular_learning3.0>cd democomponents  
C:\Users\Gabriel\Desktop\angular_learning3.0\democomponents>npm install jquery
```

```
Run 'npm audit' for details.
```

```
C:\Users\Gabriel\Desktop\angular_learning3.0\democomponents>npm install bootstrap
```

**\*\*Adding the js files references\*\***

Add them in the angular.json > scripts

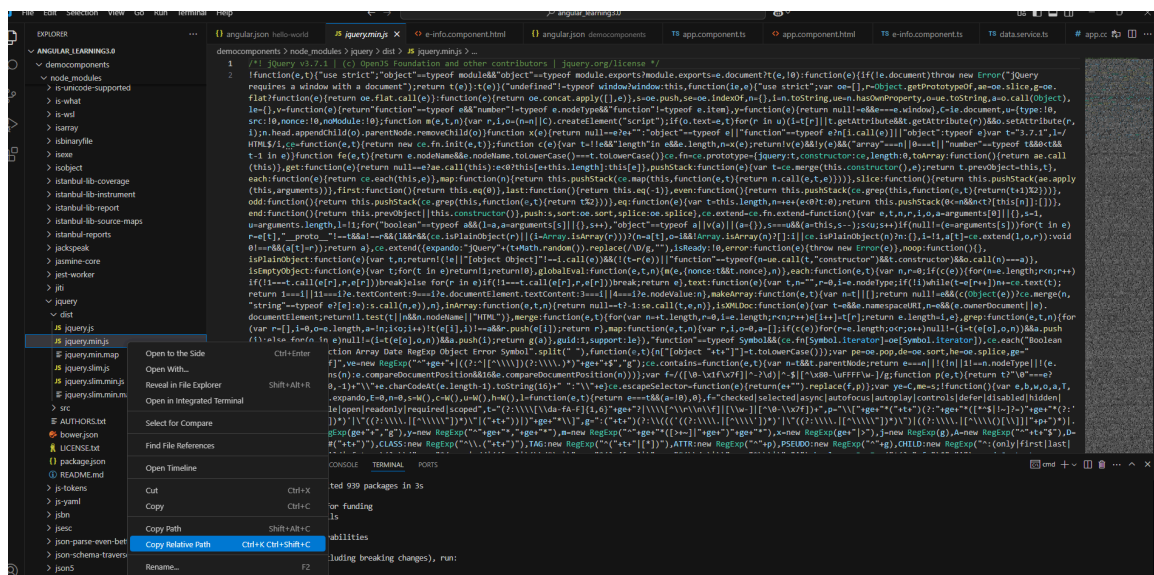
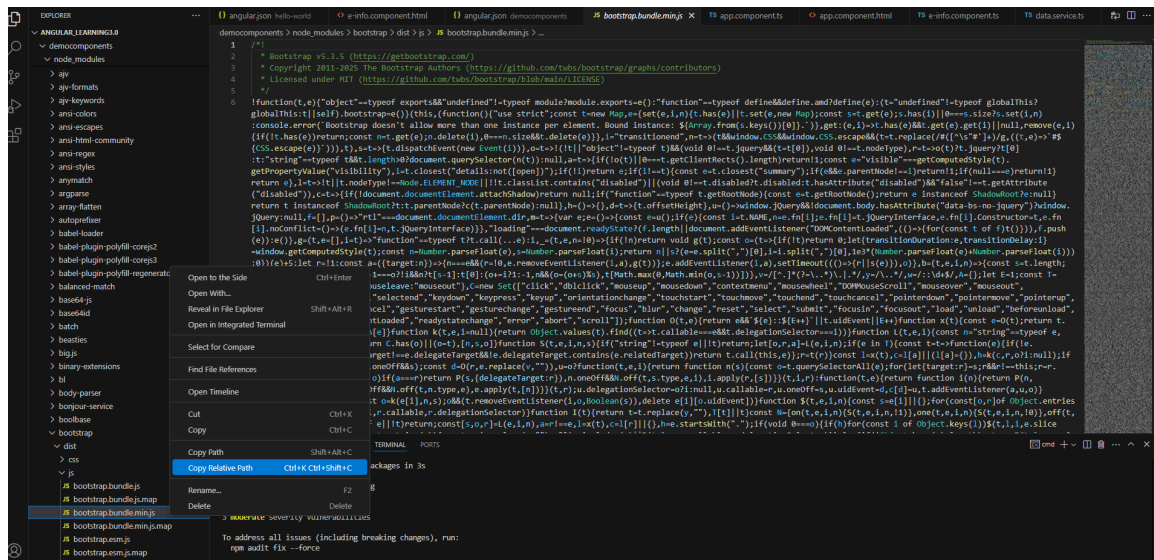
```
angular_learning3.0
angular.json hello-world e-info.component.html {} angular.json democomponents X TS app.component
democomponents > {} angular.json > {} projects > {} democomponents > {} architect > {} build > {} options > [ ] scrip
"projects": {
  "democomponents": {
    "prefix": "app",
    "architect": {
      "build": {
        "builder": "@angular-devkit/build-angular:application",
        "options": {
          "outputPath": "dist/democomponents",
          "index": "src/index.html",
          "browser": "src/main.ts",
          "polyfills": [
            "zone.js"
          ],
          "tsConfig": "tsconfig.app.json",
          "assets": [
            "src/assets",
            "src/favicon.ico"
          ],
          "styles": [
            "src/styles.css"
          ],
          "scripts": [
          ],
          "server": "src/main.server.ts",
```

To find the files, they are at folder

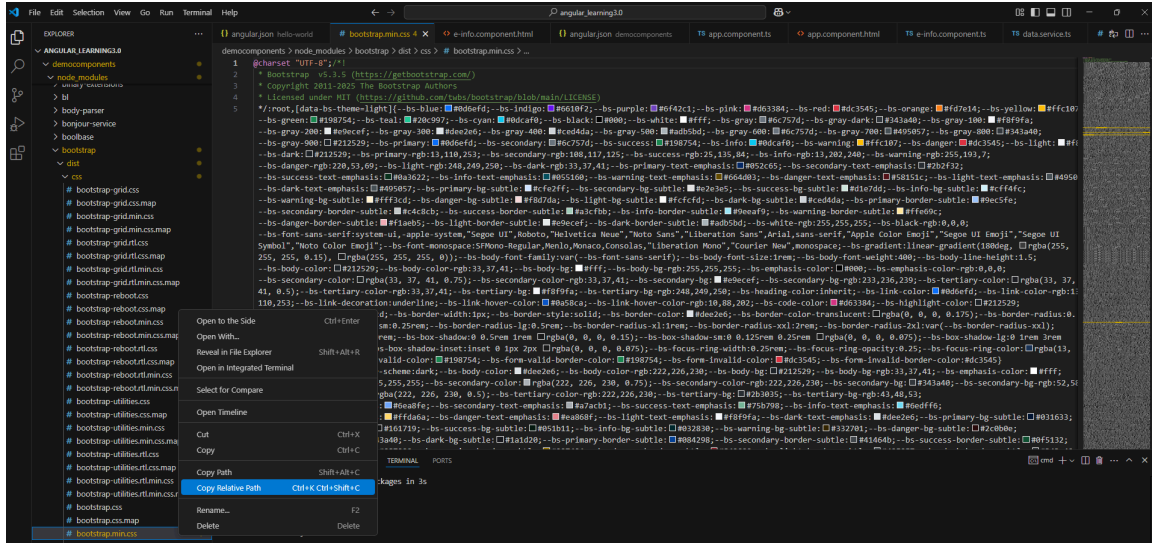
node\_modules > bootstrap > dist > js>

node\_modules > jquery > dist

node\_modules > bootstrap > dist > css







democomponents\node\_modules\bootstrap\dist\js\bootstrap.bundle.min.js

democomponents\node\_modules\jquery\dist\jquery.min.js

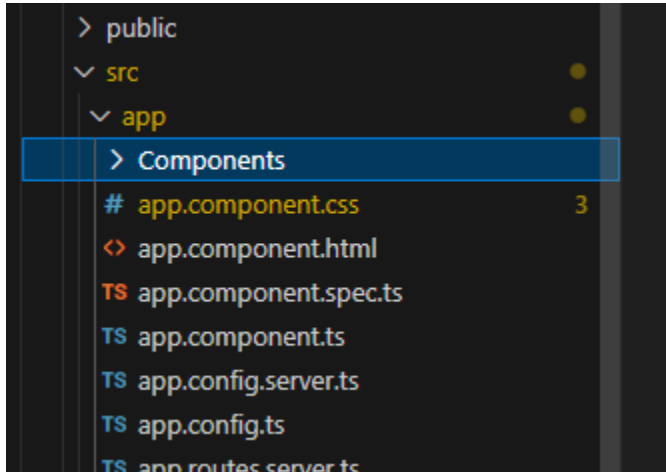
democomponents\node\_modules\bootstrap\dist\css\bootstrap.min.css

Change to forward slash

```
"scripts": [
  "democomponents/node_modules/bootstrap/dist/js/bootstrap.bundle.min.js",
  "democomponents/node_modules/jquery/dist/jquery.min.js"
]
```

```
],
"styles": [
  "src/styles.css",
  "democomponents/node_modules/bootstrap/dist/css/bootstrap.min.css"
],
"scripts": [
  "democomponents/node_modules/bootstrap/dist/js/bootstrap.bundle.min.js",
  "democomponents/node_modules/jquery/dist/jquery.min.js"
]
```

Create a folder to organize our components and code



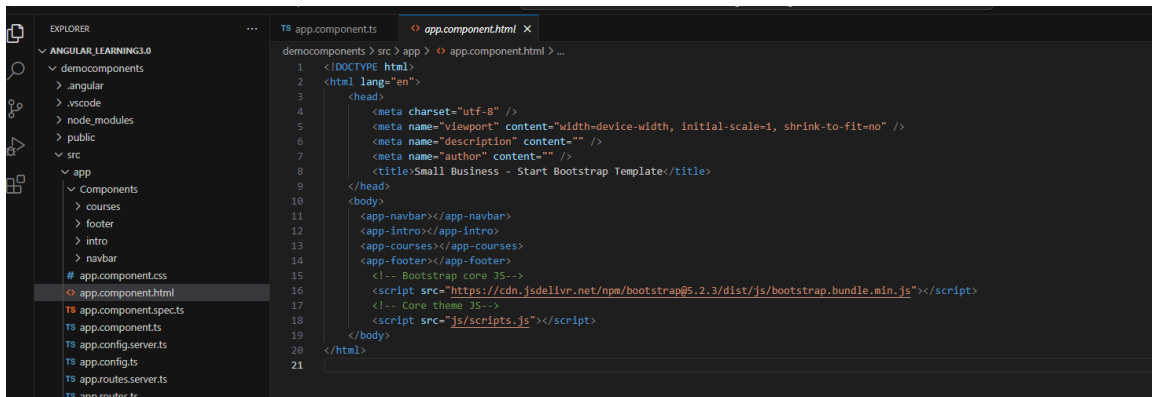
ng g c Components/navbar

ng g c Components/intro

ng g c Components/courses

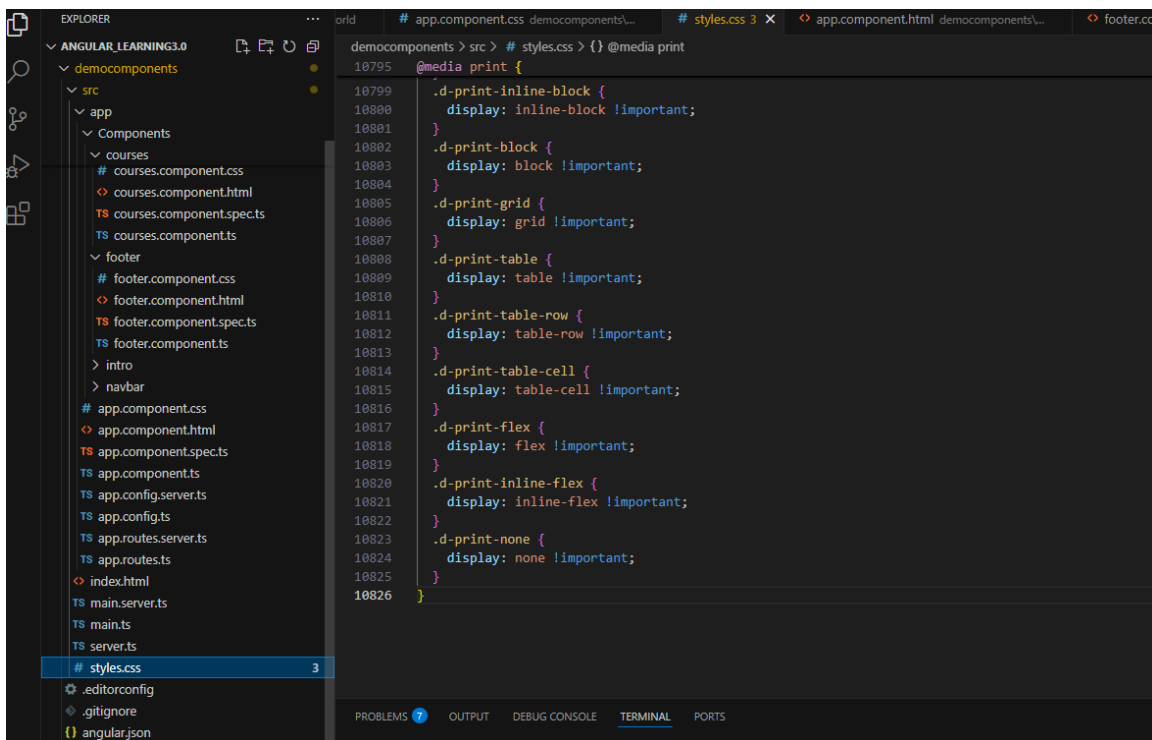
ng g c Components/footer

```
TS app.components.ts X
democomponents > src > app > TS app.components.ts > ...
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3  import { NavbarComponent } from './Components/navbar/navbar.component';
4  import { IntroComponent } from './Components/intro/intro.component';
5  import { CoursesComponent } from './Components/courses/courses.component';
6  import { FooterComponent } from './Components/footer/footer.component';
7
8
9  @Component({
10   selector: 'app-root',
11   imports: [RouterOutlet, NavbarComponent, IntroComponent, CoursesComponent, FooterComponent],
12   templateUrl: './app.component.html',
13   styleUrls: ['./app.component.css']
14 })
15 export class AppComponent {
16   title = 'democomponents';
17 }
18
```

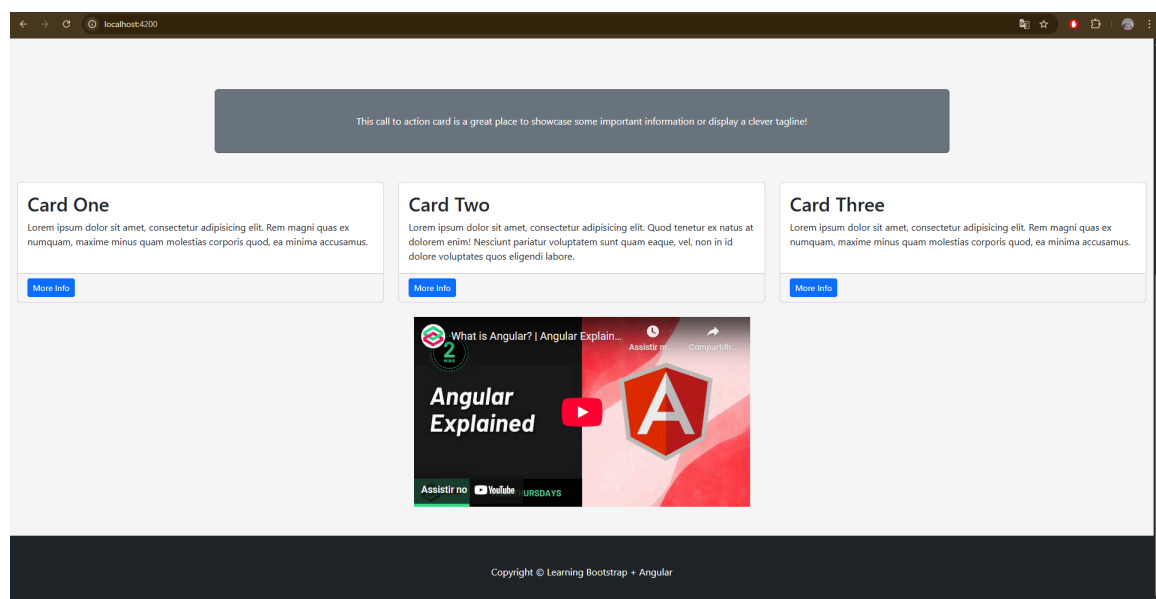
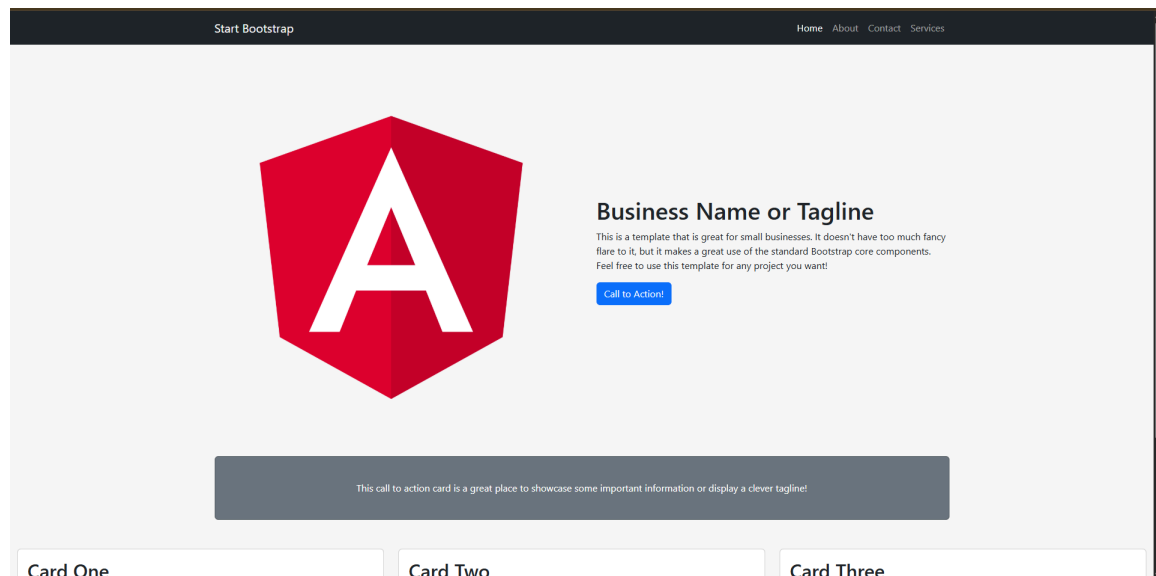


```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
6     <meta name="description" content="" />
7     <meta name="author" content="" />
8     <title>Small Business - Start Bootstrap Template</title>
9   </head>
10  <body>
11    <app-navbar></app-navbar>
12    <app-intro></app-intro>
13    <app-courses></app-courses>
14    <app-footer></app-footer>
15    <!-- Bootstrap core JS-->
16    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
17    <!-- Core theme JS-->
18    <script src="js/scripts.js"></script>
19  </body>
20 </html>
21
```

Add the style at src/style.css



```
1
2
3
```



## Angular Routing

### Basics of routing

Routing is essentially the mechanism that helps you navigate through a website or a web-application.

Especially in angular, routing is used to create single pages.

This application is loaded once and new features are just mere additions to the user interface, added dynamically.

Ex: Google, X, Facebook, Gmail...

SPAs are much faster and give a desktop-like feel to the application.

**\*\*Routing is the mechanism used to navigate into the application and load new content**

## Development

As of the latest versions of Angular (v17+), route management has changed a bit because of the new "standalone API", where components and modules can be used without directly depending on the AppModule.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

C:\Users\Gabriel\Desktop\angular_learning3.0\democomponents>ng g c login
CREATE src/app/login/login.component.html (21 bytes)
CREATE src/app/login/login.component.spec.ts (608 bytes)
CREATE src/app/login/login.component.ts (221 bytes)
CREATE src/app/login/login.component.css (0 bytes)

C:\Users\Gabriel\Desktop\angular_learning3.0\democomponents>ng g c home
CREATE src/app/home/home.component.html (20 bytes)
CREATE src/app/home/home.component.spec.ts (601 bytes)
CREATE src/app/home/home.component.ts (217 bytes)
CREATE src/app/home/home.component.css (0 bytes)
```

Separate them into home and login

```
terminal Help
home.component.html X
democomponents > src > app > home > home.component.html
1 <app-intro></app-intro>
2 <app-courses></app-courses>
3 <app-footer></app-footer>
```

```
login.component.html X
democomponents > src > app > login > login.component.html > div.container.d-flex.justify-content-center.align-items-center.vh-100
1 <div class="container d-flex justify-content-center align-items-center vh-100">
2   <div class="card p-4 shadow" style="width: 100%; max-width: 400px;">
3     <h3 class="text-center mb-3">Login</h3>
4     <form>
5       <div class="mb-3">
6         <label for="email" class="form-label">Email address</label>
7         <input type="email" class="form-control" id="email" placeholder="Enter your email">
8       </div>
9
10      <div class="mb-3">
11        <label for="password" class="form-label">Password</label>
12        <input type="password" class="form-control" id="password" placeholder="Enter your password">
13      </div>
14
15      <div class="d-grid mb-3">
16        <button type="submit" class="btn btn-primary">Login</button>
17      </div>
18
19      <p class="text-center small">
20        Don't have an account? <a href="#">Register</a>
21      </p>
22    </form>
23  </div>
24</div>
```

**\*\*Add the routes\*\***

```
import { Routes } from '@angular/router';

import { LoginComponent } from '../login/login.component';

import { HomeComponent } from '../home/home.component';

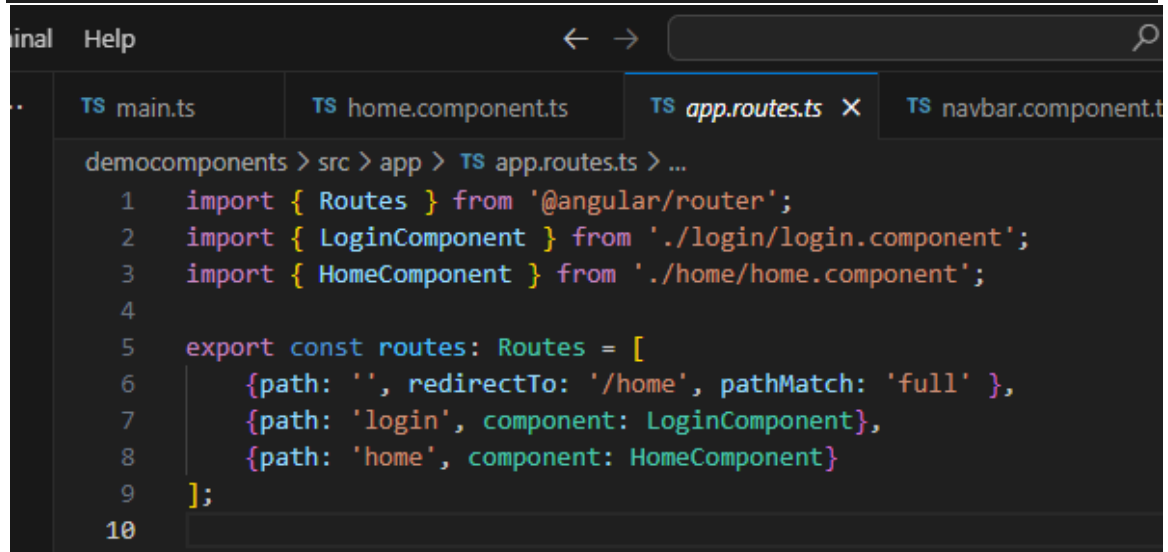
export const routes: Routes = [

  {path: '', redirectTo: '/home', pathMatch: 'full' },

  {path: 'login', component: LoginComponent},

  {path: 'home', component: HomeComponent}

];
```



The screenshot shows an IDE window with the file `app.routes.ts` selected. The code in the file is as follows:

```
democomponents > src > app > TS app.routes.ts > ...
1  import { Routes } from '@angular/router';
2  import { LoginComponent } from '../login/login.component';
3  import { HomeComponent } from '../home/home.component';
4
5  export const routes: Routes = [
6    {path: '', redirectTo: '/home', pathMatch: 'full' },
7    {path: 'login', component: LoginComponent},
8    {path: 'home', component: HomeComponent}
9  ];
10
```

**\*\*Import the RouterModule in all necessary ts files\*\***

```
TS main.ts TS home.component.ts X TS navbar.component.ts TS login.component.ts app.component.html
democomponents > src > app > home > TS home.component.ts > ...
1 import { Component } from '@angular/core';
2 import { IntroComponent } from '../Components/intro/intro.component';
3 import { CoursesComponent } from '../Components/courses/courses.component';
4 import { FooterComponent } from '../Components/footer/footer.component';
5 import { CommonModule } from '@angular/common';
6 import { RouterModule } from '@angular/router';
7
8
9 @Component({
10   selector: 'app-home',
11   imports: [IntroComponent, CoursesComponent, FooterComponent, CommonModule, RouterModule],
12   templateUrl: './home.component.html',
13   styleUrls: ['./home.component.css']
14 })
15 export class HomeComponent {
16
17 }
18
```

**\*\*Include the routes to the navbar\*\***

```
<!-- Responsive navbar-->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container px-5">
    <a class="navbar-brand" href="#!">Angular Learning</a>
    <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation"><span
class="navbar-toggler-icon"></span></button>
    <div class="collapse navbar-collapse"
id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        <li class="nav-item active"><a class="nav-link"
aria-current="page" routerLink="/home"
routerLinkActive="active">Home</a></li>
        <li class="nav-item"><a class="nav-link"
>About</a></li>
```



```

        <li class="nav-item"><a class="nav-link"
>Contact</a></li>

        <li class="nav-item"><a class="nav-link"
routerLink="/login" routerLinkActive="active">Login</a></li>

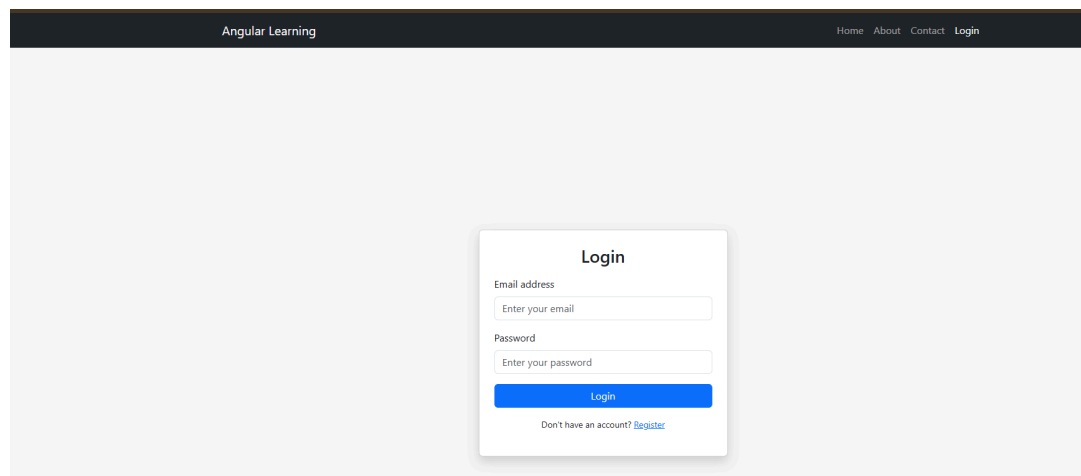
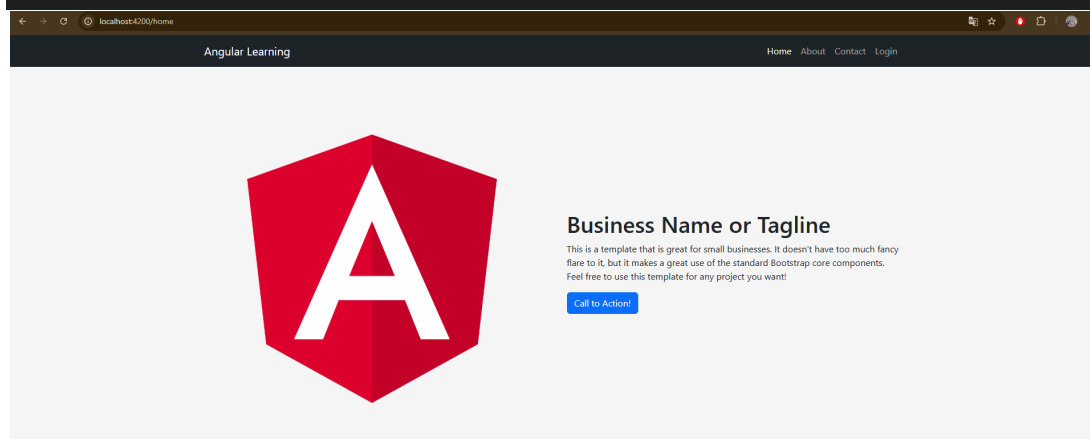
    </ul>

</div>

</div>

</nav>

```



Made with ❤️ while learning Angular with Simplilearn – Angular Basics.