Lesson 02 Demo 05
Implementing Routing Mechanisms

Objective: To implement static and dynamic routing

Tools required: Ubuntu, Visual Studio, and Node.js

Pre-requisites: Knowledge of JavaScript, Typescript, and Node.js

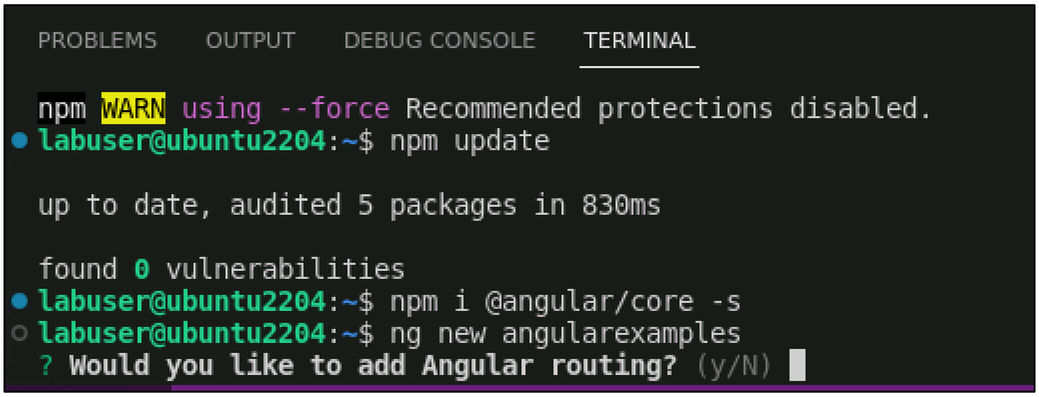
Steps to be followed:

1. Create an angular app and perform static routing
2. Perform dynamic routing

Step 1: Create an angular app

1.1 Open Visual Studio

1.2 Create a new project, using the command **ng new angularexamples** and enter yes

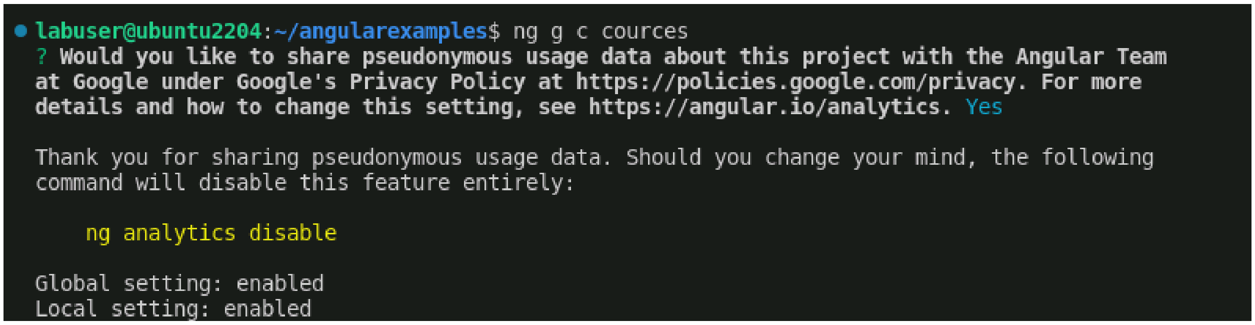


```
npm WARN using --force Recommended protections disabled.
labuser@ubuntu2204:~$ npm update

up to date, audited 5 packages in 830ms

found 0 vulnerabilities
labuser@ubuntu2204:~$ npm i @angular/core -s
labuser@ubuntu2204:~$ ng new angularexamples
? Would you like to add Angular routing? (y/N)
```

1.3 Open the project folder created. **File->open folder->angularexamples**

1.4 Now, go to the terminal and type the following command:

ng g c cources



```
● labuser@ubuntu2204:~/angularexamples$ ng g c cources
  ? Would you like to share pseudonymous usage data about this project with the Angular Team
  at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
  details and how to change this setting, see https://angular.io/analytics. Yes

  Thank you for sharing pseudonymous usage data. Should you change your mind, the following
  command will disable this feature entirely:

      ng analytics disable

  Global setting: enabled
  Local setting: enabled
```

Note: These are names of the tabs where we will be performing the routing on.

1.5 Go to **src->app->app-routing.module.ts** file and write the code below:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { CourcesComponent } from './cources/cources.component';
const routes: Routes = [
  {
    path: 'cources',
  component:CourcesComponent,  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

1.6 Go to **app.component.html** file and write the code below:

```
<h1> {{title}} - Static Routing</h1>
<a routerLink="cources">Cources</a>
```

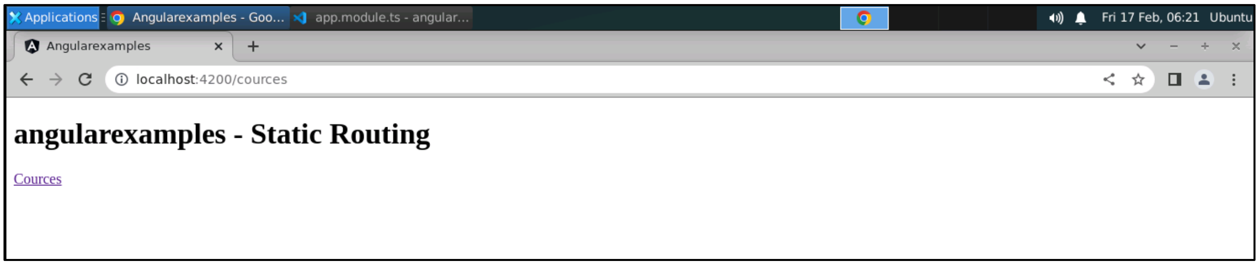
```
<br>

<router-outlet></router-outlet>
```

1.7 Go to **app.module.ts** file and see that the code should be automatically be updated as below:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { CourcesComponent } from './cources/cources.component';
@NgModule({
  declarations: [
    AppComponent,
    CourcesComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

1.8 Run it on browser or run the command: **npm run start**

Click on any of the service and you will see that it is redirected to that service, as shown below:



angularexamples - Static Routing

Cources

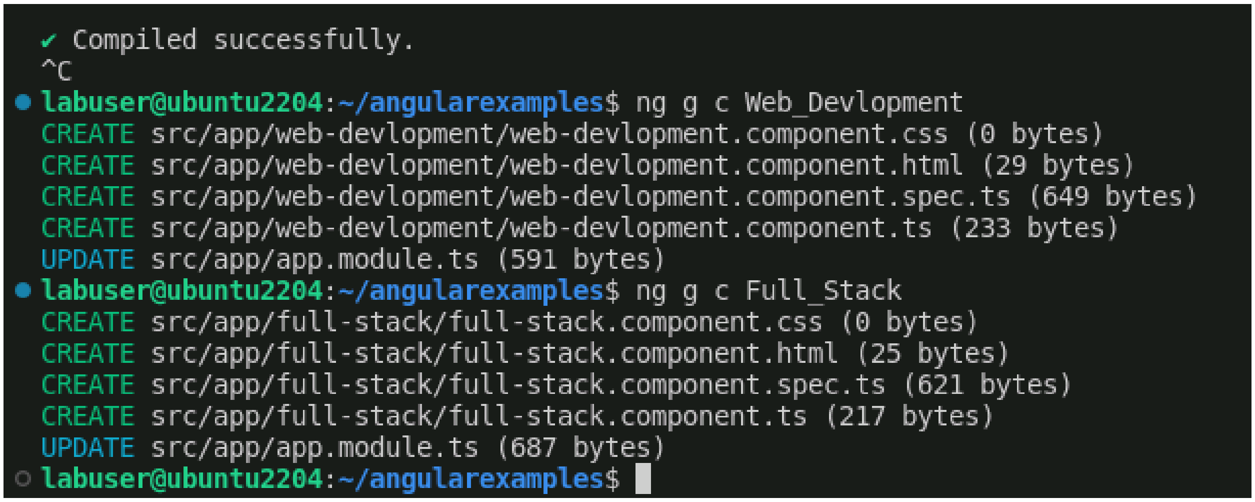
Note: The above component comes under static routing.

Step 2: Perform dynamic routing

2.1 Now, go to the terminal and type the commands:

ng g c Web_Devlopment

ng g c Full_Stack



```
✔ Compiled successfully.
^C
● labuser@ubuntu2204:~/angularexamples$ ng g c Web_Devlopment
  CREATE src/app/web-devlopment/web-devlopment.component.css (0 bytes)
  CREATE src/app/web-devlopment/web-devlopment.component.html (29 bytes)
  CREATE src/app/web-devlopment/web-devlopment.component.spec.ts (649 bytes)
  CREATE src/app/web-devlopment/web-devlopment.component.ts (233 bytes)
  UPDATE src/app/app.module.ts (591 bytes)
● labuser@ubuntu2204:~/angularexamples$ ng g c Full_Stack
  CREATE src/app/full-stack/full-stack.component.css (0 bytes)
  CREATE src/app/full-stack/full-stack.component.html (25 bytes)
  CREATE src/app/full-stack/full-stack.component.spec.ts (621 bytes)
  CREATE src/app/full-stack/full-stack.component.ts (217 bytes)
  UPDATE src/app/app.module.ts (687 bytes)
○ labuser@ubuntu2204:~/angularexamples$ 
```

2.2 Go to **app-routing.module.ts** file and replace the code as below. Here we have added new routes for demonstrations purpose:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { CourcesComponent } from './cources/cources.component';
import { WebDevlopmentComponent } from './web-devlopment/web-
devlopment.component';
import { FullStackComponent } from './full-stack/full-stack.component';

const routes: Routes = [
{
 path: 'cources',
 component:CourcesComponent,
 children:[
 {
  path: 'Web_Devlopment',
  component:WebDevlopmentComponent
 },
 {
  path: 'Full_Stack/:id',
  component:FullStackComponent
 }
 ]
}
];

@NgModule({
 imports: [RouterModule.forRoot(routes)],
 exports: [RouterModule]
})
export class AppRoutingModule { }
```

2.2 Go to **cources.component.html** file and replace the code as below:

```
<h1>Dynamic Routing</h1>
<a routerLink="Web_Devlopment">Web Devlopment Cource</a>
<br><br>
<a routerLink="Full_Stack/121">Full Stack Cource</a>
<br>
<router-outlet></router-outlet>
```

2.3 Go to **Full_Stack.component.html** file and replace the code as below:

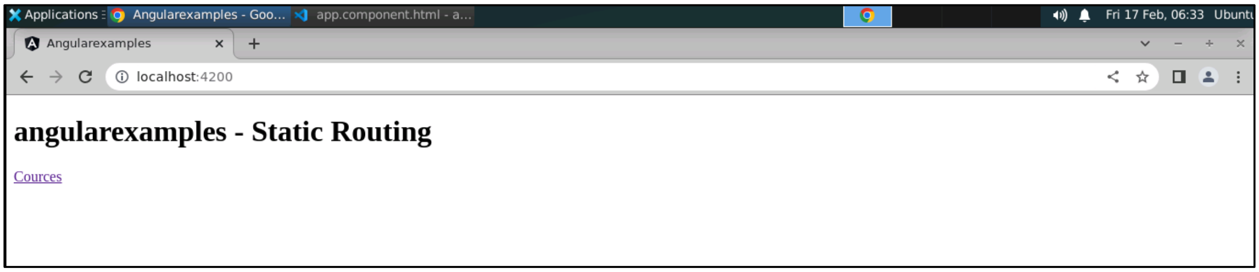
```
<p>full-stack works!</p>
<h2>ID is : {{ID}}</h2>
```

2.4 Go to **Full-stack.component.ts** file and replace the code as below:

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

@Component({
 selector: 'app-full-stack',
 templateUrl: './full-stack.component.html',
 styleUrls: ['./full-stack.component.css']
})
export class FullStackComponent implements OnInit {

 constructor(private route: ActivatedRoute) { }
```

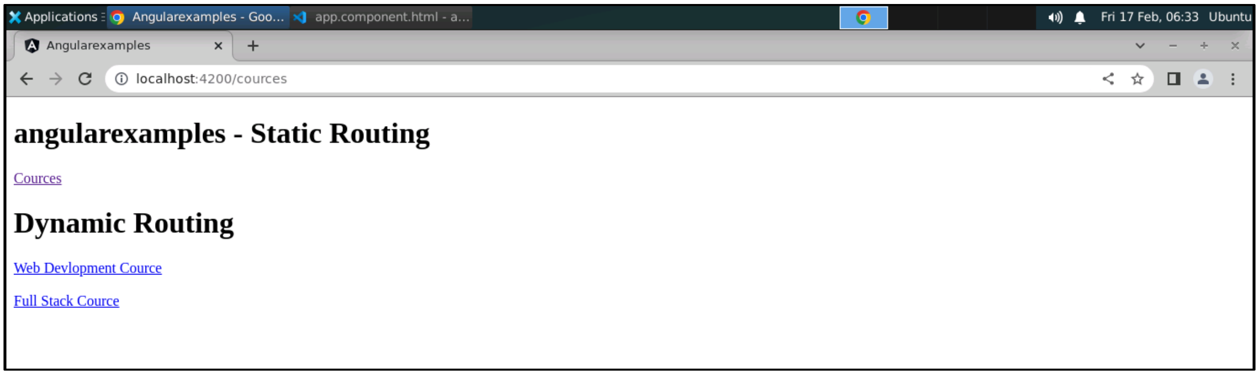
```
    ID:any;

    ngOnInit(): void {
      this.ID = this.route.snapshot.paramMap.get('id');
    }

  }
```

2.5 Save the file and run it on browser or use the command: **npm run start**



angularexamples - Static Routing

Cources



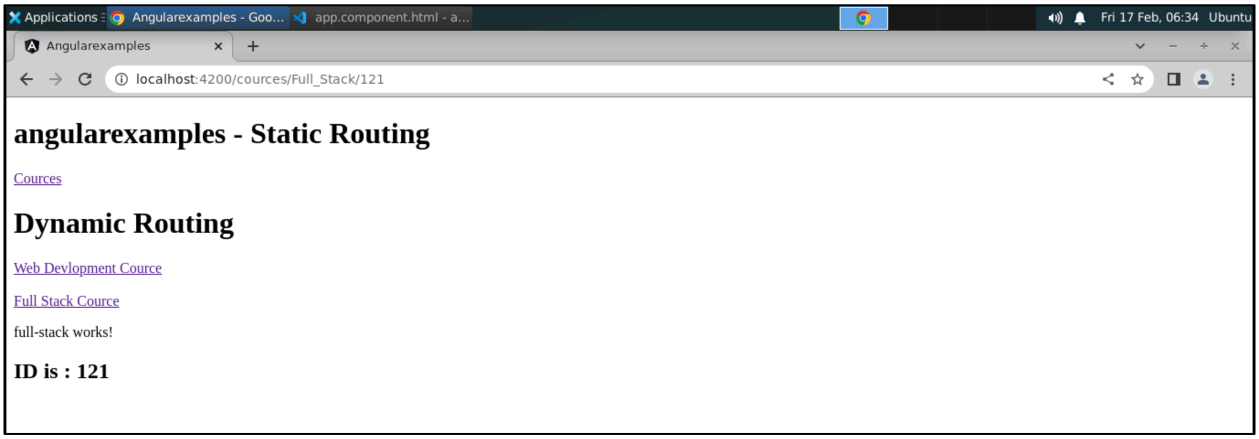
angularexamples - Static Routing

Cources

Dynamic Routing

Web Devlopment Cource

Full Stack Cource



angularexamples - Static Routing

Cources

Dynamic Routing

Web Devlopment Cource

Full Stack Cource

full-stack works!

ID is : 121

Note: Full stack come under dynamic routing as we are passing ID parameter for it. However, web development and course fall under static routing, where we are routing it to a specific place and not using an ID.