Angular Training

(Intermediate to Advanced)



Narasimha

Sr. Corporate Trainer, Mentor tnrao.trainer@gmail.com

Schedule for Angular Training

Day#	Date	Topic
Day-1	16-Nov-2023	Custom Pipes in Angular
Day-2	17-Nov-2023	Parent-Child Communication
Day-3	20-Nov-2023	Custom Directives in Angular
Day-4	21-Nov-2023	Working with Reactive Forms
Day-5	22-Nov-2023	Dependency Injection and Services in Angular
Day-6	23-Nov-2023	Http Client – Server calls in Angular
Day-7	24-Nov-2023	Routing and Security in Angular
Day-8	27-Nov-2023	Unit Testing in Angular

Duration: 8 days (2hours per day); 2pm to 4 pm; 16th Nov – 27th Nov

Day-7 Routing and Security in Angular



Narasimha

Sr. Corporate Trainer, Mentor tnrao.trainer@gmail.com

Index



- What & Why routing?
- How to setup routing?
- Router links and router outlet
- Handle Invalid Routes
- Route Parameters
- How to create secured routes
- Route Guards: CanActivate
- Programmatic Navigations



What is Routing?
Why do we need Routing in Angular?

What is SPA?

- In a Single Page Application (SPA), all of your application's functions exist in a single HTML page.
- As users access your application's features, the browser needs to render only the parts that matter to the user, instead of loading a new page.
- This pattern can significantly improve your application's user experience.



What is Routing?

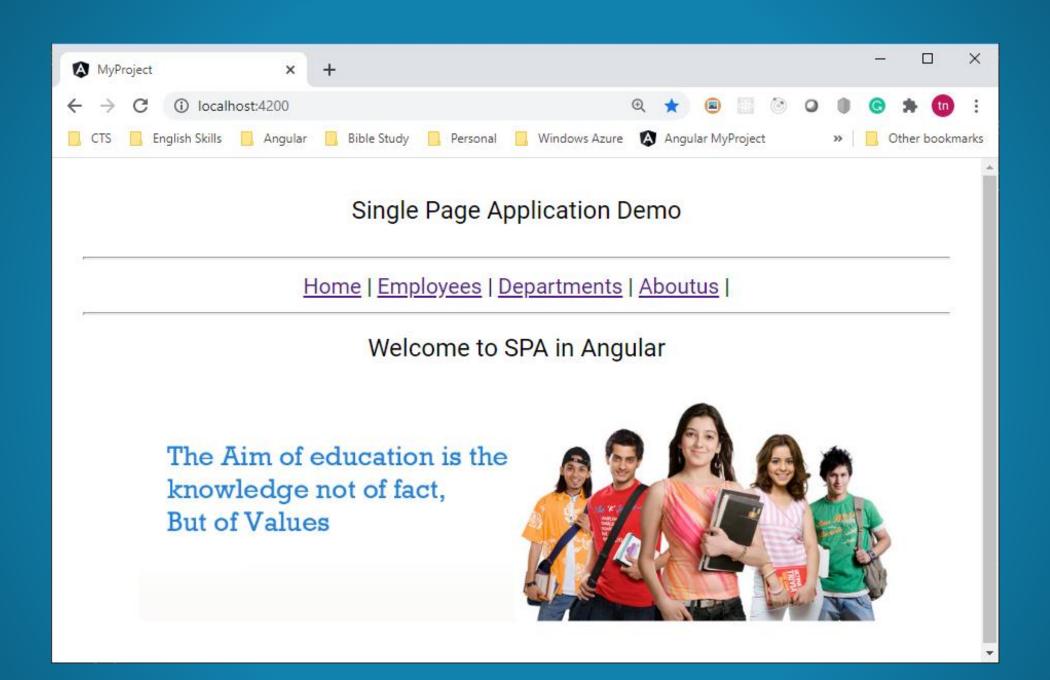
- As users perform application tasks, they need to move between the different views that you have defined.
- To handle the navigation from one view to the next, you use the Angular Router.
- The Router enables navigation by interpreting a browser URL as an instruction to change the view.

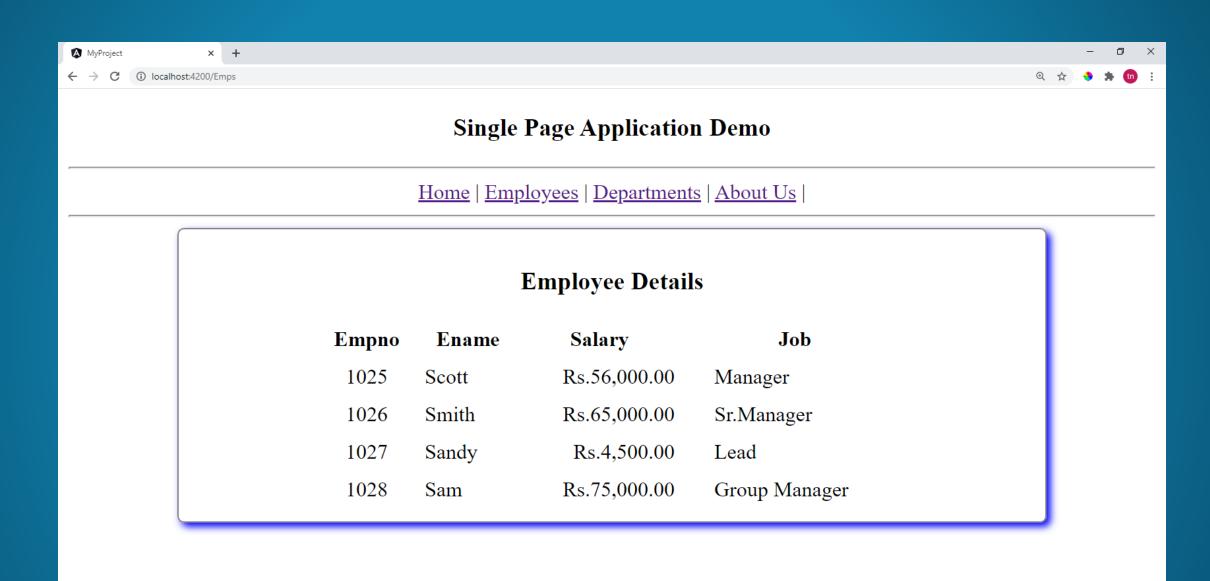


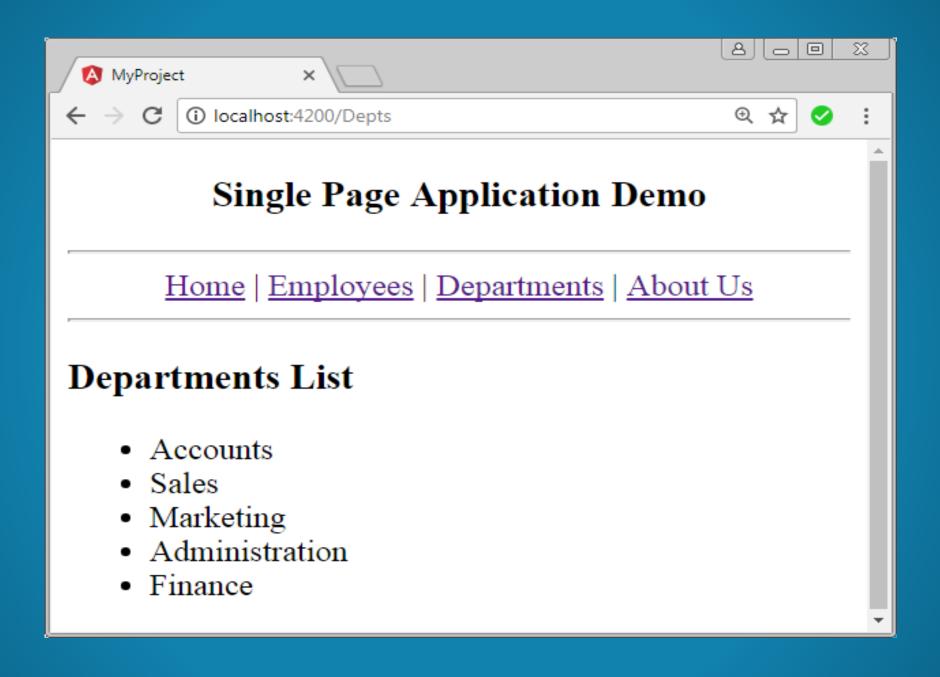
What is Routing?

- To define how users navigate through your application, you use routes.
- Add routes to define how users navigate from one part of your application to another.
- You can also configure routes to guard against unexpected or unauthorized behavior.



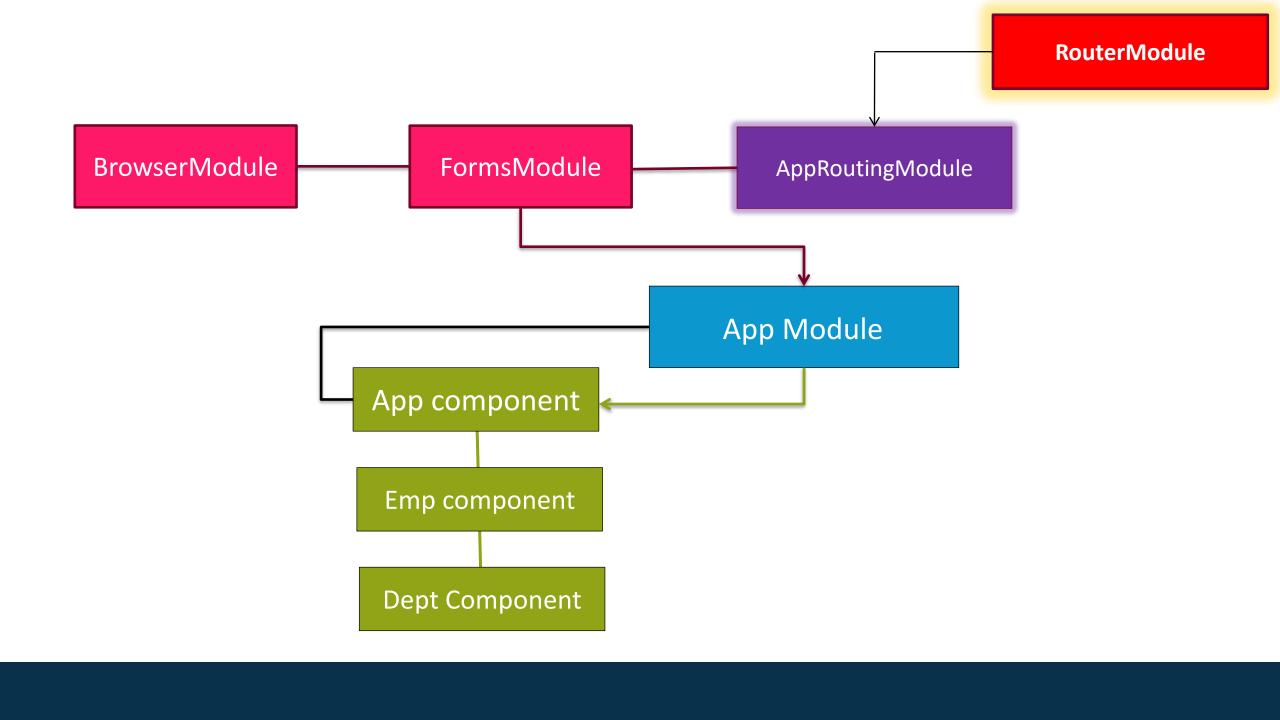








How to setup the Routing?



Configuring Routes (app.module.ts)

```
import { AppRoutingModule } from './app-routing.module';
@NgModule({
  imports: [
        BrowserModule
         FormsModule
         HTTPClientModule
        AppRoutingModule
  declarations: [...],
   bootsrap: [ AppComponent ]
}]
export class AppModule { }
```

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
const routes: Routes = [];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
export class AppRoutingModule { }
```

Configuring Routes

- 1. Prepare the routes
- 2. Register the routes
- 3. Create hyperlinks
- 4. Create placeholder

Configuring Routes

import { RouterModule, Routes } from '@angular/router';

```
const appRoutes:Routes = [
    { path : "", component : HomeComponent },
    { path : "Login", component : LoginComponent },
    { path : "Emps", component : EmpsComponent }
];
```

```
@NgModule({
    imports : [ RouterModule.forRoot(appRoutes ) ]
    })
```

Router links and router outlet

```
<a [routerLink]=" ['/'] "> Home </a>
<a [routerLink]=" ['/Emps'] "> Employees </a>
<a [routerLink]=" ['/Login'] "> Login</a>
```

<router-outlet></router-outlet>

RouterOutlet acts as a placeholder that Angular dynamically fills based on the current router state.

Practical HandsOns



Handle Invalid Routes

*Narasimha*Sr. IT Trainer/Consultant



Route Parameters

*Narasimha*Sr. IT Trainer/Consultant

Route Parameters

1. Configure the route with parameter:

```
{path : "details/:id", component: DetailsComponent}
```

2. Prepare hyperlink with parameter:

```
<a [routerLink]="['/Details', 12345]"> Details </a>
```

3. Read route parameter values in details component:

```
constructor(private activatedRouteObj:ActivatedRoute){ }
this.activatedRouteObj.snapshot.params["id"];
```



How to create secured routes?

Secured Routes – Autheticiation

- 1. AuthGurad Service
- 2. Route Guards: CanActivate
- 3. Local Storage / Session Storage
- 4. Programmatic Navigations using Router
- 5. LoginComponent



Local Storage / Session Storage

```
localStorage.setItem("key", "value");
localStorage.getItem("key");
localStorage.removeItem("key");
```

```
localStorage.setItem("uid", "scott");
localStorage.getItem("uid");
localStorage.removeItem("uid");
```



Practical HandsOns

Summary



- Routing
- Setup routing?
- Router links and router outlet
- Route Parameters
- Nested Routes
- Secured routes
- Route Guards: CanActivate
- Programmatic Navigations
- localStorage/sessionStorage

