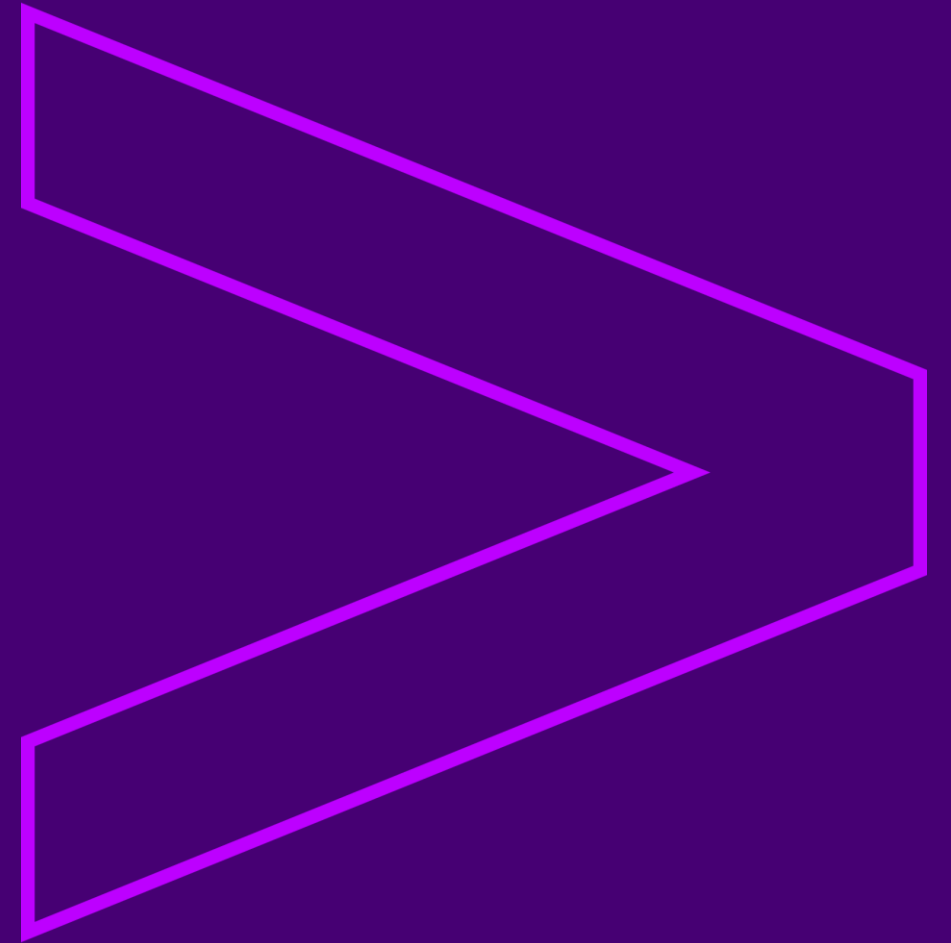


ANGULAR – ROUTING FUNDAMENTALS



MODULE OBJECTIVES

At the end of this module, you should be able to:

- Implement Routing in Angular
- Configure Routes in a SPA
- Set the default route
- Use Router directives - RouterOutlet, RouterLink & RouterLinkActive



AGENDA

- ▶ Understanding Routing
- ▶ Configuring the Routes
- ▶ Setting the default route
- ▶ Use of base href
- ▶ Router directives - RouterOutlet, RouterLink & RouterLinkActive.
- ▶ Routing Demo

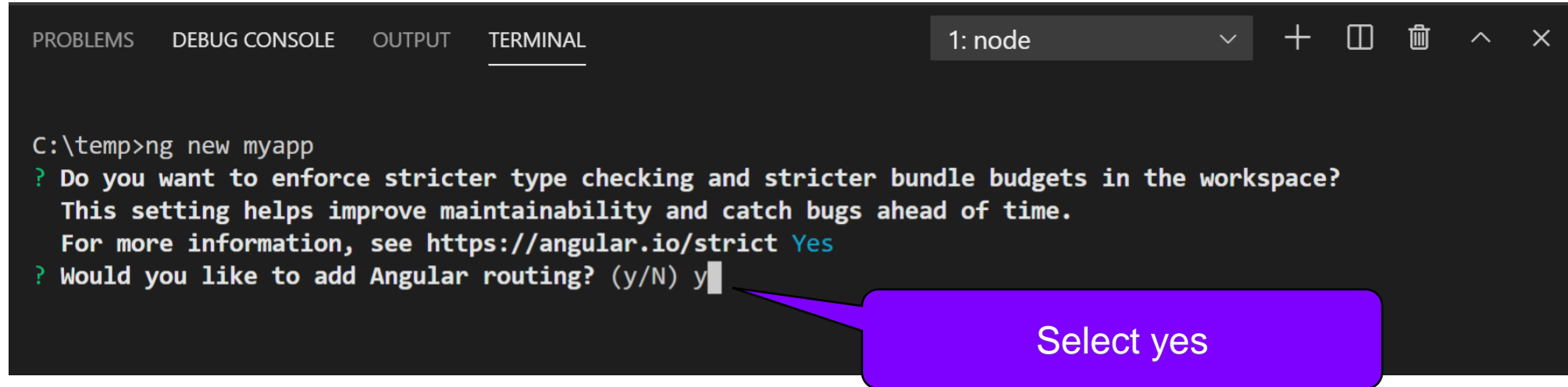


Routing in Angular

- Routing enables navigation from one view to the next as users perform application tasks.
- The Angular Router can interpret a browser URL as an instruction to navigate to a required view component.
- You can navigate imperatively when the user clicks a link/button, selects from a drop box etc
- We can pass optional parameters to the supporting view component that help it decide what specific content to present.
- Angular router logs activity in the browser's history so the back and forward buttons work as well.

Enabling routing in an Angular app

- While creating an angular app using Angular CLI, you will be prompted if routing support is required



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  1: node  +  [ ]  [X]  ^  X

C:\temp>ng new myapp
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? (y/N) y
```

- This will generate an angular NgModule named **AppRoutingModule** where routes are to be configured

CLI-generated AppRoutingModuleModule

- The default AppRoutingModuleModule generated by Angular CLI imports **Routes** and **RouterModule**
- It sets up a routes array to define the routes

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router'; // CLI imports router

const routes: Routes = []; // sets up routes constant where you define your routes

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

List of imports

To define the routes here

CLI-generated AppModule

- CLI also adds AppRoutingModuleModule to the AppModule's imports array

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppRoutingModuleModule } from './app-routing.module'; // CLI imports AppRoutingModuleModule
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModuleModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

AppRoutingModule added to the
AppModule's imports array

Understanding Routing

- To understand Angular Routing, let us create an Angular Application which has the below components
 - HomeComponent,
 - EmployeesListComponent
 - DepartmentListComponent
 - ProductsListComponent
- A navigational menu will be displayed. Based on the link clicked by user, the view will be generated by the corresponding components
- This avoids reaching out to the server for loading a new page

Configuring the Routes – Step 1

Step 1: Create the required components

```
ng g c home  
ng g c employees-list  
ng g c department-list  
ng g c products-list
```

Configuring the Routes – Step 2

Step 2: Import the new components into the AppRoutingModuleModule

```
import { AppComponent } from './app.component';  
import { EmployeesListComponent } from './employees-list/employees-list.component';  
import { DepartmentListComponent } from './department-list/department-list.component';  
import { ProductsListComponent } from './products-list/products-list.component';  
import { HomeComponent } from './home/home.component';
```

Configuring the Routes – Step 3

Step 3: Define an array of route configurations in the AppRoutingModuleModule

```
const routes: Routes=[  
  {path:'home', component:HomeComponent},  
  {path:'employees', component:EmployeesListComponent},  
  {path:'departments', component:DepartmentListComponent},  
  {path:'products', component:ProductsListComponent},  
  {path: '', redirectTo: '/home', pathMatch: 'full' },  
  {path: '**', component:HomeComponent}  
]
```

Specify the URL path and the component associated with the route

Static routes

Default route

Wildcard route if no match found

Note:

The '**' path in the last route is a wildcard. The router will select this route if the requested URL doesn't match any routes defined earlier. Useful for displaying a "404 - Not Found" page or redirecting to another route.

ROUTES Array

- Routes array has the following properties:

Property	Description
path	path is a string that uses the route matcher DSL.
pathMatch	pathMatch is a string that specifies the matching strategy.
matcher	matcher defines a custom strategy for path matching and supersedes path and pathMatch.
component	component is a component type.
redirectTo	redirectTo is the url fragment which will replace the current matched segment.
outlet	outlet is the name of the outlet the component should be placed into.
canActivate	canActivate is an array of DI tokens used to look up CanActivate handlers. See CanActivate for more info.
canActivateChild	canActivateChild is an array of DI tokens used to look up CanActivateChild handlers. See CanActivateChild for more info.
canDeactivate	canDeactivate is an array of DI tokens used to look up CanDeactivate handlers. See CanDeactivate for more info.
canLoad	canLoad is an array of DI tokens used to look up CanLoad handlers. See CanLoad for more info.
data	data is additional data provided to the component via ActivatedRoute.
resolve	resolve is a map of DI tokens used to look up data resolvers. See Resolve for more info.
runGuardsAndResolvers	runGuardsAndResolvers defines when guards and resolvers will be run. By default they run only when the matrix parameters of the route change. When set to paramsOrQueryParamsChange they will also run when query params change. And when set to always, they will run every time.
children	children is an array of child route definitions.
loadChildren	loadChildren is a reference to lazy loaded child routes. See LoadChildren for more info.

The default route

- When the application launches, the initial URL in the browser bar is something like:
<http://localhost:4200>
- A default route is added that translates the empty path (") to the desired default path (/home).
- It requires a pathMatch property to tell the router how to match a URL to the path of a route. The router throws an error if you don't.

```
const routes: Routes=[  
  {path:'home', component:HomeComponent},  
  {path:'employees', component:EmployeesListComponent},  
  {path:'departments', component:DepartmentListComponent},  
  {path:'products', component:ProductsListComponent},  
  {path: "", redirectTo: '/home', pathMatch: 'full' },  
  {path: '**', component:HomeComponent}  
]
```

Wildcard routes

- A wildcard route is selected when the requested URL doesn't match any of the configured routes
- A wildcard route must be the last route as it matches any URL

```
const routes: Routes=[  
  {path:'home', component:HomeComponent},  
  {path:'employees', component:EmployeesListComponent},  
  {path:'departments', component:DepartmentListComponent},  
  {path:'products', component:ProductsListComponent},  
  {path: "", redirectTo: '/home', pathMatch: 'full' },  
  {path: '**', component:HomeComponent}  
]
```

Note : Mostly the wildcard route sends the user to the **PageNotFoundComponent**

base href

- All routing applications should add a <base> element to the index.html as the first child in the <head> tag to tell the router how to compose navigation URLs.
- By default, this is added in the index.html in CLI-generated angular apps

Example: index.html

```
<!doctype html>
<html lang="en">
<head>
<title>RoutingDemo1</title>
<base href="/">
</head>
<body>
<app-root></app-root>
</body>
</html>
```

Router directives

- The Angular Router module provides the below Directives for use in the template
 - RouterLink
 - RouterLinkActive
 - RouterOutlet

1. RouterLink

- Once the routes are defined, use the **routerLink** directive to add routes in the app
- The `<a>` tag must contain the routerLink attribute

app.component.html

```
<h1 align="center">{{title}}</h1>
<nav align="center">
  <a routerLink="/home" routerLinkActive="active">Home</a>
  <a routerLink="/employees" routerLinkActive="active">Employees List</a>
  <a routerLink="/departments" routerLinkActive="active">Departments List</a>
  <a routerLink="/products" routerLinkActive="active">Products List</a>
</nav>
```

2. RouterLinkActive

- The RouterLinkActive directive allows to set CSS classes when the link is clicked
- The template expression to the right of the equals (=) contains a space-delimited string of CSS classes that the Router will add when this link is active (and remove when the link is inactive).

app.component.html

```
<h1 align="center">{{title}}</h1>
<nav align="center">
  <a routerLink="/home" routerLinkActive="active">Home</a>
  <a routerLink="/employees" routerLinkActive="active">Employees List</a>
  <a routerLink="/departments" routerLinkActive="active">Departments List</a>
  <a routerLink="/products" routerLinkActive="active">Products List</a>
</nav>
```

3. RouterOutlet

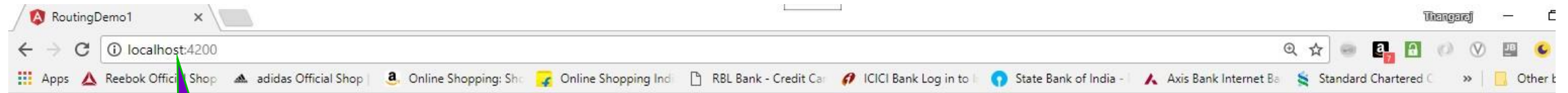
- The RouterOutlet is a directive from the router library that marks the spot in the template where the router should display the view of the selected component
- The routed view will be rendered in the <router-outlet>

app.component.html

```
<h1 align="center">{{title}}</h1>
<nav align="center">
  <a routerLink="/home" routerLinkActive="active">Home</a>
  <a routerLink="/employees" routerLinkActive="active">Employees List</a>
  <a routerLink="/departments" routerLinkActive="active">Departments List</a>
  <a routerLink="/products" routerLinkActive="active">Products List</a>
</nav>

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

Routing Demo - Outputs



Routing Demo Application

[Home](#) [Employees List](#) [Departments List](#) [Products List](#)

Welcome

ABC Company

Bangalore

http://localhost:4200

Routing Demo - Outputs



Routing Demo Application

[Home](#) [Employees List](#) [Departments List](#) [Products List](#)

Welcome
ABC Company
Bangalore

Default
and also
when
home
linked is
Clicked

http://localhost:4200/home



Routing Demo - Outputs

Routing Demo Application

[Home](#) [Employees List](#) [Departments List](#) [Products List](#)

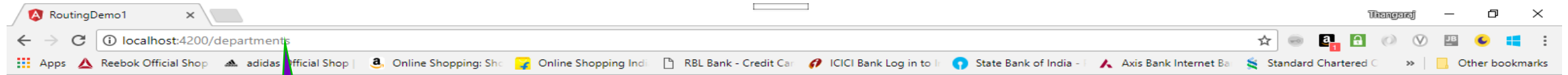
Employees List

Employee No	Employee Name	Employee Age	Employee Salary
1231	Thangaraj	21	12345.11
1232	Sindhu	22	34512.22
1233	Snigdha	23	12343.33
1234	Bhagya	24	12344.44
1235	Bhavya	25	12345.55
1236	Pavana	26	12346.66
1237	Seema	27	12347.77
1238	Karthik	28	12348.88
1239	Prashantha	29	12349.99
1240	Manoj	30	12340

Clicked

http://localhost:4200/employees

Routing Demo - Outputs



Routing Demo Application

[Home](#) [Employees List](#) [Departments List](#) [Products List](#)

Departments List

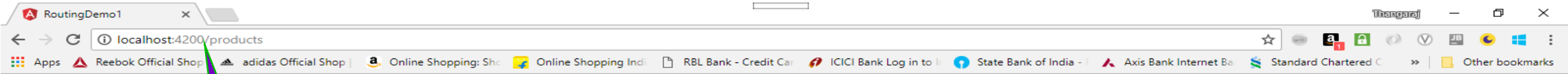
Department No	Department Name	Department Location
11	BigData	Chennai
12	BigData	Chennai
13	Angular	Pune
14	C++	Gurgaon
15	SAP	Noida
16	Testing	Bangalore
17	SFDC	Hyderabad
18	Oracle	Kolkata
19	Avanade	Trivendrum
20	Analytics	Chicago

Clicked

http://localhost:4200/departments



Routing Demo - Outputs



Routing Demo Application

[Home](#) [Employees List](#) [Departments List](#) [Products List](#)

Products List

Product No	Product Name	Product Price
111	IPhone	Rs. 45000
121	IPad	Rs. 34500
131	IPod	Rs. 32222
141	Moto G	Rs. 35678
151	Moto X	Rs. 43434
161	Moto E	Rs. 67676
171	Moto Z	Rs. 23123
181	Samsung Note 3	Rs. 43211
191	Samsung Note 4	Rs. 76433
201	Samsung Note 5	Rs. 37890

Clicked

http://localhost:4200/products



Routing Demo - Outputs



Routing Demo Application

[Home](#) [Employees List](#) [Departments List](#) [Products List](#)

Welcome

ABC Company

Bangalore

<http://localhost:4200/productssdfsfs>

Unhandled route, so
defaults to Home Page



MODULE SUMMARY

Now, you should be able to:

- Implement Routing in Angular
- Configure Routes in a SPA
- Set the default route
- Use Router directives - RouterOutlet, RouterLink & RouterLinkActive



QUESTIONS



THANK YOU