



deti universidade de aveiro
departamento de electrónica,
telecomunicações e informática

Technologies and Web Programming

Angular Framework



Angular Framework

Routing in Angular

Angular – Routing (i)



- The browser is a familiar model of application navigation:
 - Enter a URL in the address bar and the browser navigates to a corresponding page;
 - Click links on the page and the browser navigates to a new page;
 - Click the browser's back and forward buttons and the browser navigates backward and forward through the history of pages you've seen.

Angular – Routing (ii)



- The Angular Router is based on this model:
 - It can interpret a browser URL as an instruction to navigate to a client-generated view.
 - It can pass optional parameters along to the supporting view component that help it decide what specific content to present.
 - You can bind the router to links on a page and it will navigate to the appropriate application view when the user clicks a link.

Angular – Routing (iii)



- The Angular router is based on this model:
 - You can navigate imperatively when the user clicks a button, selects from a drop box, or in response to some other stimulus from any source.
 - And the router logs activity in the browser's history journal so the back and forward buttons work as well.

Routing – Developing (i)



- Angular's, best practice, is to load and configure the router in a separate, top-level module that is dedicated to routing.
- By convention, this module resides in the “app.routes.ts” file, in the “src/app” folder, which exports a “Routes” typed constant.

Routing – Developing (v)



- Adding Routes
 - Routes tell the router which view to display when a user clicks a link or pastes a URL into the browser address bar.
 - A typical Angular Route has two properties:
 - **path**: a string that matches the URL in the browser address bar.
 - **component**: the component that the router should create when navigating to this route.

Routing – Developing (vi)



- Adding Routes (cont.)
 - Example – navigate to Authors component when the URL is something like localhost:4200/authors.
 - Import Authors component so you can reference it in a Route. Then define an array of routes with a single route to that component.

```
TS app.routes.ts ×  
1   import { Routes } from '@angular/router';  
2   import { Authors } from './authors/authors';  
3  
4   export const routes: Routes = [ Show usages new *  
5   { path: 'authors', component: Authors }  
6   ];
```

Routing – Developing (viii)



- Adding Routes (cont.)
 - Add RouterOutlet and RouterLink to “app.ts”, like this:

```
TS app.ts x <> app.html
1 import { Component } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { RouterLink, RouterOutlet } from '@angular/router';
4 import { Authors } from './authors/authors';
5
6
7 @Component({ Show usages ⓘ Helder Zagalo *
8   selector: 'app-root',
9   imports: [CommonModule, RouterOutlet, RouterLink, Authors],
10  templateUrl: './app.html',
11  styleUrls: ['./app.css'],
12 })
13 export class App {
14   title : string = 'My Books';
15 }
```

Routing – Developing (ix)



- Adding Routes (cont.)
 - After importing RouterOutlet, Angular is ready to route all defined routes.
 - Add its selector to “app.html” like this:

```
<> app.html x
1   <h1>{{ title }}</h1>
2   <router-outlet></router-outlet>
```

- Now, you can already try the URL “http://localhost:4200/authors”

Routing – Developing (x)



- Adding Routes (cont.)
 - You can also add navigations link to navigate between home and authors:

```
<> app.component.html x
1  <h1><a routerLink="">{{ title }}</a></h1>
2
3  <nav>
4    <a routerLink="/authors">Authors</a>
5  </nav>
6
7  <router-outlet></router-outlet>
```

- Using routerLink attribute is a better way than href attribute. Try to use both to see the differences.



Angular Framework

Multiple Views in Angular

Multi-Views



- Adding multiple views:
 - Adding a Top panel.
 - Adding the ability to navigate between Authors and Top views.
 - When users click an author in either view, it navigates to a Details view of the selected author.
 - When users click a deep link in an email, it opens the Details view for a particular author.

Multi-Views – Top view (i)



- Create Top component:
 - Command line: ng generate component top
- In “top.html” put the following html code:

```
<> top.html x
1   <h2>Top Authors</h2>
2
3   <div class="grid grid-pad">
4     @for (au of authors; track au.id) {
5       <a class="col-1-4">
6         <div class="module author">
7           <h4>{{au.name}}</h4>
8         </div>
9       </a>
10    }
11 </div>
```

Multi-Views – Overview view (ii)



- Code “top.ts” in the same way as “authors.ts”, with light difference:

```
ts top.ts x
1  import { Component } from '@angular/core';
2  import { RouterLink } from '@angular/router';
3  import { Author } from '../author';
4  import { AUTHORS } from '../authorslist';
5
6
7  @Component({ Show usages
8  selector: 'app-top',
9  imports: [RouterLink],
10 templateUrl: './top.html',
11 styleUrls: ['./top.css'],
12 })
13 export class Top {
14   authors: Author[];
15
16   constructor() { no usages
17     this.authors = AUTHORS.slice( start: 0, end: 4);
18   }
19 }
```

Multi-Views – Overview view (iii)



- Replace the Top Component CSS file with the one given in e-learning “top.css”.

```
top.css x
1 |
2 [class*='col-'] {
3   float: left;
4   padding-right: 20px;
5   padding-bottom: 20px;
6 }
7 [class*='col-']:last-of-type {
8   padding-right: 0;
9 }
10 a {
11   text-decoration: none;
12 }
13 *, *:after, *:before {
14   -webkit-box-sizing: border-box;
15   -moz-box-sizing: border-box;
16   box-sizing: border-box;
17 }
```

Multi-Views – Overview view (iv)



- Add the Top route:

```
TS app.routes.ts ×
1   import { Routes } from '@angular/router';
2   import { Authors } from './authors/authors';
3   import { Top } from './top/top';
4
5   export const routes: Routes = [ Show usages new *
6     { path: 'authors', component: Authors },
7     { path: 'top', component: Top },
8   ];
```

Multi-Views – Overview view (v)



- Add links to the main component:

<> app.component.html x

```
1 <h1>{{ title }}</h1>
2
3 <nav>
4   <a routerLink="">Home</a>
5   <a routerLink="/top">Top</a>
6   <a routerLink="/authors">Authors</a>
7 </nav>
8
9 <router-outlet></router-outlet>
```

- Give it a try ...

Multi-Views – Overview view (vi)



- Replace the App CSS file with the one given in e-learning “app.css”:

```
app.css x
1   nav {
2     display: flex;
3     gap: 0.25rem;
4     padding: 0.5rem;
5     border-bottom: 1px solid #e5e7eb;
6   }
7
8   nav a {
9     padding: 0.4rem 0.85rem;
10    font-size: 14px;
11    color: #6b7280;
12    text-decoration: none;
13    border-radius: 6px;
14    transition: color 0.15s, background 0.15s;
15  }
16
17  nav a:hover {
18    color: #111827;
19    background: #f3f4f6;
20  }
```

Multi-Views – Details view (i)



- Create AuthorDetails component:
 - Command line: `ng generate component author-details`
- Cut and paste the HTML code relative to details from “authors.html” to “author-details.html” and modify as below:

```
<> author-details.html x
1  @if (author) {
2    <h2>Information on {{author.name | uppercase}}</h2>
3    <div>Id: {{author.id}}</div>
4    <div>
5      <label>
6        Id:
7        <input [(ngModel)]="author.id" readonly />
8      </label>
9    </div>
10   <div>
11     <label>
12       Name:
13       <input [(ngModel)]="author.name" placeholder="name" />
14     </label>
15   </div>
16   <div>
17     <label>
18       Email:
19       <input [(ngModel)]="author.email" placeholder="email" />
20     </label>
21   </div>
22 }
```

Multi-Views – Details view (ii)



- Replace the AuthorDetails component CSS file with the one given in e-learning “author-details.css”.

```
author-details.css x
1 |
2 label {
3   display: inline-block;
4   width: 3em;
5   margin: .5em 0;
6   color: #607D8B;
7   font-weight: bold;
8 }
9 input {
10  height: 2em;
11  font-size: 1em;
12  padding-left: .4em;
13 }
14 button {
15  margin-top: 20px;
16  font-family: Arial;
17  background-color: #eee;
18  border: none;
19  padding: 5px 10px;
20  border-radius: 4px;
21  cursor: pointer; cursor: hand;
22 }
```

Multi-Views – Details view (iii)



- Add the author details route:

```
ts app.routes.ts x
1   import { Routes } from '@angular/router';
2   import { Authors } from './authors/authors';
3   import { Top } from './top/top';
4   import { AuthorDetails } from './author-details/author-details';
5
6   export const routes: Routes = [ Show usages new *
7     { path: 'authors', component: Authors },
8     { path: 'top', component: Top },
9     { path: 'authordetails/:id', component: AuthorDetails }
10  ];
```

Multi-Views – Details view (iv)



- Modify “author-details.ts” as follows:

```
TS author-details.ts ×
1   import { Component } from '@angular/core';
2   import {FormsModule} from "@angular/forms";
3   import {UpperCasePipe, Location} from "@angular/common";
4   import { Author } from '../author';
5   import { ActivatedRoute } from '@angular/router';
6   import { AUTHORS } from '../authorslist';
7
8   @Component({ Show usages
9     selector: 'app-author-details',
10    imports: [FormsModule, UpperCasePipe],
11    templateUrl: './author-details.html',
12    styleUrls: ['./author-details.css'],
13  })
14  export class AuthorDetails {
15    author: Author | undefined;
16  }
```

Multi-Views – Details view (v)



- Modifying “author-detailst.ts” – extract id parameter from route and find right author.

```
16
17   constructor(private route: ActivatedRoute, private location: Location) no usages
18   {
19     this.author = this.getAuthor();
20   }
21
22   getAuthor(): Author|undefined { Show usages
23     let id: any = this.route.snapshot.params['id'];
24     if (id == null)
25       return undefined;
26     id = +id;
27     return AUTHORS.find(au: Author => au.id === id);
28   }
```

Multi-Views – Details view (vi)



- Modifying “author-details.ts” – add goBack method

```
goBack() : void {  
  this.location.back();  
}
```

- In “author-details.html” add button.

```
15 <div>  
16   <label>Email:  
17   <input [(ngModel)]="author.email" placeholder="email" />  
18   </label>  
19 </div>  
20 <div>  
21   <button (click)="goBack()">Go Back</button>  
22 </div>  
23 </div>
```

Multi-Views – Details view (vii)



- Update “authors.html” like this:

```
<> authors.html x
1   <h2>Authors</h2>
2   <ul class="authors">
3     @for (au of authors; track au.id) {
4       <li>
5         <a routerLink="/authordetails/{{au.id}}">
6           <span class="badge">{{ au.id }}</span> {{au.name}}
7         </a>
8       </li>
9     }
10  </ul>
```

- Authors component doesn't use “selectedAuthor” anymore, so its code can be deleted from “authors.ts”.

Multi-Views – Details view (viii)



- Update “top.html” like this:

```
<> top.html x
1  <h2>Top Authors</h2>
2
3  <div class="grid grid-pad">
4    @for (au of authors; track au.id) {
5      <a class="col-1-4">
6        <div class="module author" routerLink="/authordetails/{{au.id}}">
7          <h4>{{au.name}}</h4>
8        </div>
9      </a>
10   }
11 </div>
```



Angular Framework

Bootstrap and Angular

Bootstrap & Angular



- One easy way to use Bootstrap in Angular is to import all needed files in “index.html” file:

```
<> index.html ×
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>Books</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-scale=1">
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
10   <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.2/font/bootstrap-icons.css" rel="stylesheet">
11 </head>
12 <body>
13   <div class="container">
14     <app-root></app-root>
15   </div>
16   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
17 </body>
18 </html>
```

Bootstrap & Angular



- Example:

The screenshot shows a web browser window with the address bar displaying `localhost:4200/overview`. The page has a dark navigation bar with the text "The Best Books" and a list of links: "Home", "Overview", "Authors", "Publishers", and "Books". To the right of these links is a search input field with the placeholder text "Search" and a green "Search" button. Below the navigation bar, the main content area is titled "Top Authors" and contains four blue buttons with white text: "Fernando Pessoa", "J. K. Rowling", "Stephen King", and "Arthur Conan Doyle". At the bottom of the page, there is a footer with the text "© The Best Books".