

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 AuthorsComponent when the URL is something like localhost:4200/authors.
 - Import AuthorsComponent 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 x
1  import { Routes } from '@angular/router';
2  import {AuthorsComponent} from './authors/authors.component';
3
   2 usages  new *
4  export const routes: Routes = [
5  Ⓢ↑      {path: 'authors', component: AuthorsComponent}
6  ];
```


Routing – Developing (viii)



- Adding Routes (cont.)
 - If needed, add RouterOutlet to “app.component.ts”, like this:

```
TS app.component.ts x
1  import { Component } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { RouterOutlet } from '@angular/router';
4  import { AuthorsComponent } from './authors/authors.component';
5
6  @Component({
7    selector: 'app-root',
8    standalone: true,
9    imports: [CommonModule, RouterOutlet, AuthorsComponent],
10   templateUrl: './app.component.html',
11   styleUrls: ['./app.component.css']
12 })
13 export class AppComponent {
14   title: string = 'My Books';
15 }
```

Routing – Developing (ix)



- Adding Routes (cont.)
 - If RouterOutlet available is imported, than Angular is ready to route all defined routes.
 - Add its selector to “app.component.html” like this:

```
<> app.component.html ×
1 <h1>{{ title }}</h1>
2
3 <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.component.html” put the following html code:

`<> top.component.html` x

```
1 <h2>Top Authors</h2>
2 <div class="grid grid-pad">
3   <a *ngFor="let author of authors" class="col-1-4">
4     <div class="module author">
5       <h4>{{author.name}}</h4>
6     </div>
7   </a>
8 </div>
```

Multi-Views – Overview view (ii)



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

```
ts top.component.ts x
1  import { Component } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { Author } from "../author";
4  import { AUTHORS } from "../authorslist";
5
6  5+ usages  new *
7  @Component({
8    selector: 'app-top',
9    standalone: true,
10   imports: [CommonModule],
11   templateUrl: './top.component.html',
12   styleUrls: ['./top.component.css']
13 })
14 export class TopComponent {
15   authors: Author[];
16
17   no usages  new *
18   constructor() {
19     this.authors = AUTHORS.slice(0,4);
20   }
21 }
```

Multi-Views – Overview view (iii)



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

```
top.component.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:

```
top.component.css  TS app.routes.ts ×  
  
1  import { Routes } from '@angular/router';  
2  import {AuthorsComponent} from "../authors/authors.component";  
3  import {TopComponent} from "../top/top.component";  
4  
   2 usages  new *  
5  export const routes: Routes = [  
6    {path: 'authors', component: AuthorsComponent},  
7    {path: 'top', component: TopComponent}  
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 – Details view (i)



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

<> author-details.component.html x

```
1 <div *ngIf="author">
2   <h2>Information on {{selectedAuthor.name | uppercase}}</h2>
3   <div><span>Num: </span>{{selectedAuthor.num}}</div>
4   <div>
5     <label>Num:
6       <input [(ngModel)]="selectedAuthor.num" readonly />
7     </label>
8     <label>Name:
9       <input [(ngModel)]="selectedAuthor.name" placeholder="name" />
10    </label>
11  </div>
12  <div>
13    <label>Email:
14      <input [(ngModel)]="selectedAuthor.email" placeholder="email" />
15    </label>
16  </div>
17 </div>
```

Multi-Views – Details view (ii)



- In “author-details.component.ts” file put following lines:

```
author-details.component.ts x
1  import { Component, OnInit } from '@angular/core';
2  import { Author } from '../author';
3
4  @Component({
5    selector: 'app-author-details',
6    templateUrl: './author-details.component.html',
7    styleUrls: ['./author-details.component.css']
8  })
9  export class AuthorDetailsComponent implements OnInit {
10    author: Author;
11
12    constructor() {}
```

Multi-Views – Details view (iii)



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

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

Multi-Views – Details view (iv)



- Add the author details route:

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

Multi-Views – Details view (v)



- Update “authors.component.html” like this:

`<> authors.component.html` ×

```
1
2 <h2>Authors</h2>
3 <ul class="authors">
4   <li *ngFor="let author of authors">
5     <a routerLink="/authordetails/{{author.num}}">
6       <span class="badge">{{ author.num }}</span> {{author.name}}
7     </a>
8   </li>
9 </ul>
```

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

Multi-Views – Details view (vi)



- Update “top.component.html” like this:

<> top.component.html x

```
1 <h3>Top Authors</h3>
2 <div class="grid grid-pad">
3   <a *ngFor="let author of authors" class="col-1-4" routerLink="/authordetails/{{author.num}}">
4     <div class="module author">
5       <h4>{{author.name}}</h4>
6     </div>
7   </a>
8 </div>
```


Multi-Views – Details view (vii)



- Get the route that led to the view. Add the following:

```
TS author-details.component.ts ×
1  import {Component, Input } from '@angular/core';
2  import { CommonModule, Location } from '@angular/common';
3  import {FormsModule, ReactiveFormsModule} from "@angular/forms";
4  import {Author} from "../author";
5  import {ActivatedRoute} from "@angular/router";
6  import {AUTHORS} from "../authorslist";
7
8  5+ usages  new *
9  @Component({
10     selector: 'app-author-details',
11     standalone: true,
12     imports: [CommonModule, ReactiveFormsModule, FormsModule],
13     templateUrl: '../author-details.component.html',
14     styleUrls: ['../author-details.component.css']
15 })
16 export class AuthorDetailsComponent {
17     @Input() author: Author | undefined = undefined;
18
19     no usages  new *
20     constructor(private route: ActivatedRoute, private location: Location) {
21     }
22 }
```

Multi-Views – Details view (viii)



- Extract the num parameter from route.
- Add following code:

```
18 constructor(private route: ActivatedRoute, private location: Location) {
19     this.getAuthor();
20 }
21
22 1 usage new *
23 getAuthor() : void {
24     let num : any = this.route.snapshot paramMap.get("num");
25     if (num != null)
26         num = +num;
27     else
28         return;
29     this.author = AUTHORS.find(author : Author => author.num === num);
30 }
```

Multi-Views – Details view (ix)



- Add a goBack button to the view:
 - In “author-details.component.ts” file add the following:

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

- In “author-details.component.html” file add the following:

```
12   <div>  
13     <label>Email:  
14       <input [(ngModel)]="author.email" placeholder="email" />  
15     </label>  
16   </div>  
17   <button (click)="goBack()">Go Back</button>  
18 </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 x
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:

