

ChatGPT●

## Bài 4: Angular Routing (Điều hướng giữa các Component)

### 1. Giới thiệu Angular Routing

Routing trong Angular giúp chuyển đổi giữa các component dựa trên URL mà người dùng truy cập. Hệ thống này hoạt động giống như cách một trang web truyền thống điều hướng giữa các trang.

Trong Angular, chúng ta sử dụng **Angular Router** để quản lý điều hướng giữa các component mà không cần tải lại trang.

### 2. Cài đặt Angular Router

Mặc định khi tạo một dự án Angular mới, Angular Router đã được cài đặt sẵn. Nếu bạn đang làm việc với một dự án cũ mà chưa có Router, bạn có thể cài đặt nó bằng lệnh:

```
sh
```

```
ng add @angular/router
```

Sau đó, bạn cần import **RouterModule** vào ứng dụng của mình.

---

### 3. Tạo Dự án Mẫu

---

Chúng ta sẽ tạo một ứng dụng Angular với các trang sau:

- Trang chủ (Home)
- Trang Giới thiệu (About)
- Trang Liên hệ (Contact)
- Trang Không tìm thấy (Not Found)

#### Bước 1: Tạo một dự án mới

---

```
sh
```

```
ng new angular-routing-demo  
cd angular-routing-demo  
ng serve
```

#### Bước 2: Tạo các component

---

Chúng ta sẽ tạo bốn component để điều hướng giữa chúng.

sh

```
ng generate component components/home
ng generate component components/about
ng generate component components/contact
ng generate component components/not-found
```

---

## 4. Cấu hình Routing

---

Mở file `app-routing.module.ts` và thiết lập các tuyến đường như sau:

`app-routing.module.ts`

---

typescript

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from '../components/home/home.component';
import { AboutComponent } from '../components/about/about.component';
import { ContactComponent } from '../components/contact/contact.component';
import { NotFoundComponent } from '../components/not-found/not-found.component';

const routes: Routes = [
  { path: '', component: HomeComponent }, // Trang chủ
  { path: 'about', component: AboutComponent }, // Trang Giới thiệu
  { path: 'contact', component: ContactComponent }, // Trang Liên hệ
  { path: '**', component: NotFoundComponent } // Trang không tìm thấy
];

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

Giải thích:

- `path: ''`: Điều hướng mặc định khi vào trang chủ.
- `path: 'about'`: Định tuyến đến trang About.
- `path: 'contact'`: Định tuyến đến trang Contact.
- `path: '**'`: Nếu không khớp với bất kỳ đường dẫn nào, sẽ hiển thị trang Not Found.

---

## 5. Thêm `<router-outlet>` vào `app.component.html`

---

Angular Router sẽ hiển thị component tương ứng với đường dẫn tại vị trí này.

`app.component.html`

---

html

```
<nav>
  <ul>
    <li><a routerLink="/">Home</a></li>
    <li><a routerLink="/about">About</a></li>
    <li><a routerLink="/contact">Contact</a></li>
  </ul>
</nav>

<!-- Nơi hiển thị component tương ứng với route -->
<router-outlet></router-outlet>
```

Ở đây:

- `<a routerLink="/">` là cách tạo liên kết điều hướng trong Angular.
- `<router-outlet>` là nơi component sẽ được hiển thị khi điều hướng.

---

## 6. Cập nhật Giao diện của Các Component

---

Chúng ta sẽ chỉnh sửa từng component để dễ phân biệt.

`home.component.html`

---

html

```
<h2>Welcome to the Home Page</h2>
<p>This is the home page of our Angular application.</p>
```

### about.component.html

---

html

```
<h2>About Us</h2>
<p>Learn more about our company and team.</p>
```

### contact.component.html

---

html

```
<h2>Contact Us</h2>
<p>Feel free to reach out via email or phone.</p>
```

### not-found.component.html

---

```
html
```

```
<h2>404 - Page Not Found</h2>
<p>Sorry, the page you are looking for does not exist.</p>
<a routerLink="/">Go to Home</a>
```

---

## 7. Chạy Ứng Dụng

---

Bây giờ, chạy lệnh:

```
sh
```

```
ng serve
```

Mở trình duyệt tại <http://localhost:4200/> và thử điều hướng giữa các trang.

---

## 8. Điều Hướng Bằng Code (Programmatic Navigation)

---

Bên cạnh `routerLink`, chúng ta cũng có thể điều hướng bằng code trong TypeScript.

Ví dụ: Thêm nút điều hướng trong `home.component.ts`:

```
home.component.ts
```

---

typescript

```
import { Component } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent {
  constructor(private router: Router) {}

  goToAbout() {
    this.router.navigate(['/about']);
  }
}
```

## Cập nhật `home.component.html`

---

html

```
<h2>Welcome to the Home Page</h2>
<p>This is the home page of our Angular application.</p>
<button (click)="goToAbout()">Go to About</button>
```

---

## 9. Sử dụng Route Parameters

---

Chúng ta có thể truyền tham số vào URL, ví dụ:

## Cập nhật `app-routing.module.ts`

---

typescript

```
const routes: Routes = [  
  { path: 'profile/:id', component: ProfileComponent }  
];
```

## Tạo **ProfileComponent**

---

sh

ng generate component components/profile

## Cập nhật **profile.component.ts**

---



typescript

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

@Component({
  selector: 'app-profile',
  templateUrl: './profile.component.html',
  styleUrls: ['./profile.component.css']
})
export class ProfileComponent implements OnInit {
  userId: string | null = '';

  constructor(private route: ActivatedRoute) {}

  ngOnInit() {
    this.userId = this.route.snapshot.paramMap.get('id');
  }
}
```

## Cập nhật `profile.component.html`

---

html

```
<h2>Profile Page</h2>
<p>Welcome, user with ID: {{ userId }}</p>
```

Thử truy cập <http://localhost:4200/profile/123>, bạn sẽ thấy ID `123` được hiển thị.

---

## 10. Kết Luận

---

- Chúng ta đã tìm hiểu cách thiết lập Angular Router để điều hướng giữa các component.
- Học cách sử dụng `<router-outlet>` và `<a routerLink="">` để tạo liên kết.
- Cách điều hướng bằng TypeScript (`router.navigate()`).

- Cách truyền tham số qua URL.

**Ở bài tiếp theo, chúng ta sẽ học về Reactive Forms trong Angular! 🚀**