**Tìm hiểu về Angular 2**

**1. Cơ chế One way binding**

- Là một cơ chế data binding một chiều từ View → Model hoặc từ Model → View

- Cú pháp:

+ Trong model, ta khai báo một biến mang giá trị cần binding sang View:

public var\_name = value;

+ Trong View để lấy dữ liệu được binding từ Model sang ta dùng cú pháp: {{var\_name}}

- Property binding: Là kiểu binding dữ liệu cho các thuộc tính của một thẻ hay cặp thẻ nào đó.

- Cú pháp:

+ Trong model ta khai báo một biến mang giá trị cần binding sang View:

public var\_name = value;

+ Trong View để lấy được giá trị của thuộc tính đó ta dùng cú pháp:

[name\_property] = “var\_name”

**2. Binding class và style cho element**

*\* Binding class:*

- Là cơ chế binding một class từ CSS để áp dụng các thuộc tính của class đó vào View.

- Cú pháp khai báo:

+ Trước tiên phải khởi tạo các class cùng với những thuộc tính muốn sử dụng trong CSS.

+ Trong các element ta thêm cú pháp: [class.class\_name] = “boolean\_var”

+ Trong model khai báo biến boolean\_var với cú pháp: public boolean\_var = true ( true nếu muốn áp dụng css của class class\_name, false nếu không muốn áp dụng css của class đó)

*\* Style cho element:*

- Là kiểu áp dụng một giá trị cho thuộc tính cụ thể nào đó cho element bất kỳ.

- Cú pháp: [style.color]= “apply?’style\_1’:’style\_2’”

- Trong đó: apply là biến kiểu logic được khởi tạo trong model, style\_1 và style\_2 là giá trị cho thuộc tính color của element

**3. Cách sử dụng Routing.**

- Là cách để điều hướng các liên kết trang web.

- Cách sử dụng:

- Tại file index.html cần có thẻ <base href="/"> để chỉ ra index là route gốc.

- Trong “app.mudule.ts” ta thêm thư viện Routing vào: import { RouterModule } from '@angular/router';

- Tại @ngModule (trong file app.module.ts) ta khai báo thêm các Route trong phần import, cú pháp khai báo như sau: RouterModule.forRoot([

{ path: '', component: HomeComponent },

{ path: 'page1', component: Page1Component },

{ path: 'page2', component: Page1Component }

])

trong đó HomeComponet, Page1Component, Page1Component là những Component có trong project và cần điều hướng đến, “page1, page2, … ” là những đường dẫn trên thanh địa chỉ.

- Trong app.component.html cần thêm cặp thẻ <router-outlet></router-outlet> để Angular nhận diện ra vùng thay đổi khi gọi đến Route tương ứng.

- Thêm các lrouterLink tương ứng khi muốn gọi đến các Route đó:

<li><a routerLink="">Home</a></li>

<li><a routerLink="page1">Page 1</a></li>

<li><a routerLink="page2">Page 2</a></li>

\*Lưu ý: Các routerLink phải trùng với các path đã khai báo ở RouteModule trong file “app.module.ts”

- Vd: (File source đi kèm)

- Bắt Route ngoại lệ (404 Notfound):

- Cách sử dụng:

- Trong phần khai báo danh sách các Route, ta khai báo thêm một path ở cuối cùng với cú pháp: { path: '\*\*', component: NotfoundComponent} trong đó NotfoundComponent là component là tương ứng khi không có Route nào trùng với những Route đã khai báo.

- Redirect Route: Là cú pháp để chuyển hướng Route tự động từ một Route bất kì

- Cú pháp: { path: '', redirectTo: 'page1', pathMatch:'full' }

**5. Cách gửi và nhận parameter trên URL**

- Là cách truyền và nhận tham số có trên thanh địa chỉ.

- Cách sử dụng:

- Khai báo thêm Route mới với cú pháp : { path: 'detail/:id', component: DetailComponent }

- Tại trang ta muốn truyền tham số đi ta cần khai báo routerLink theo cú pháp:

[routerLink]="['/detail', parameter]" trong đó “detail” là một router mới được khai báo trong ngModule(app.module.ts), parameter là tham số ta truyền vào.

- Khi đó trên URL có dạng: /detail/parameter

- Để nhận được tham số “parameter” tại component xử lý của “deatail” ta cần khai báo thuộc tính tương ứng để lấy parameter trên địa chỉ và 1 thuộc tính Subscription

- Khởi tạo constructor với tham số đầu vào có kiểu dữ liệu là Router và ActivatedRoute.

- Bên trong hàm khởi tạo ta xử lý parameter: this.subscription = this.activatedRoute.params.subscribe(params=>{

this.id = params['id'];

\* Lưu ý: Tại component xử lý ta cần import các thư viện ActivatedRoute và Subscription (cú pháp trong demo).