

Exercise (Instructions): Custom Attribute Directives

Objectives and Outcomes

In this exercise you will learn to create a new custom attribute directive and make use of it within a component view. At the end of this exercise you will be able to:

- Use Angular-CLI to create and add a new custom directive to your Angular application
- Design a new custom attribute directive and apply it to elements within a template in your component

Adding a New Directive

- Create a new folder named *directives* within the app folder of your project.
- Using Angular-CLI add a new directive named highlight to your application as follows:

```
1 ng g directive directives/highlight
```

- Update highlight.directive.ts file as follows:

```
1 import { Directive, ElementRef, Renderer2, HostListener } from '@angular/core';
2 . . .
3
4
5 constructor(private el: ElementRef,
6   private renderer: Renderer2) { }
7
8 @HostListener('mouseenter') onMouseEnter() {
9   this.renderer.addClass(this.el.nativeElement, 'highlight');
10 }
11
12 @HostListener('mouseleave') onMouseLeave() {
13   this.renderer.removeClass(this.el.nativeElement, 'highlight');
14 }
15
16 . . .
```

- Open styles.scss and add a new scss class to it as follows:

```
1 .highlight {  
2   background-color: $background-pale;  
3   border: 1px solid $primary-color-dark;  
4   z-index: 1;  
5   transform: scale(1.01);  
6 }
```

- Now open menu.component.html and add the new directive as follows:

```
1 . . .  
2     <md-grid-tile *ngFor="let dish of dishes" [routerLink]="['/dishdetail',  
3     dish.id]" appHighlight>
```

- Save all the changes and do a Git commit with the message "Directives".

Conclusions

In this exercise you learnt to add a new directive to your Angular application using Angular-CLI and then design a custom attribute directive and apply it to an element within a template.

Mark as completed

