**Section13 Dynamic Component in Angular**

**Notes: -**

**1-Dynamic component is used to create and initialize component at runtime**

**Features: -**

**1-Loaded programmatically (it’s the easy scenario because you are using built in tools)**

**A-Like using the \*ngIf : component embedded via selector**

**\*ngIf controls whether component is added to DOM**

**2-Dynamic Component Loader**

**Component created and added to DOM via code (imperatively)**

**Component is managed and added by developer**

**By using \*NgIf**

**<div class="backdrop" (click)="onClose()">**

**</div>**

**<div class="alert-box">**

**<p>{{message}}</p>**

**<div class="alert-box-actions">**

**<button class="btn btn-primary" (click)="onClose()">Close</button>**

**</div></div>**

**import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';**

**@Component({**

**selector: 'app-alert',**

**templateUrl: './alert.component.html',**

**styleUrls: ['./alert.component.css'],})**

**export class AlertComponent implements OnInit {**

**@Input() message: string = '';**

**@Output() close = new EventEmitter<void>();**

**constructor() {}**

**ngOnInit(): void {}**

**onClose(){**

**this.close.emit();}}**

**//on the auth.component.html we apply the output event emitter as below**

**<app-alert [message]="error" \*ngIf="error" (close)="onHandleError()"></app-alert>**

**onHandleError(){**

**this.error = null;}**

**By using Dynamic Component**

**Notes: -**

**1-we will create alert component dynamically through the code instead of using \*ngIf**

**2-in angular 9 or higher you don’t face any problem after create component and attach to container factory and attach to the directive to ng-template**

**(in earlier than Angular 9 you have to append entryCOmponents in the app.module)**

**//it means the component that used on the running dynamic creation**

**entryComponents:[AlertComponent]**

**3-we have two parts of Dynamic component**

**A-Container Factory: which used to create component at running time**

**B-Container reference: which used to host new component created**

**steps: -**

**<div class="backdrop" (click)="onClose()"></div>**

**<div class="alert-box">**

**<p>{{message}}</p>**

**<div class="alert-box-actions">**

**<button class="btn btn-primary" (click)="onClose()">Close</button>**

**</div></div>**

**//on the code behind of Alert Component**

**import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';**

**@Component({**

**selector: 'app-alert',**

**templateUrl: './alert.component.html',**

**styleUrls: ['./alert.component.css'],})**

**export class AlertComponent implements OnInit {**

**@Input() message: string = '';**

**@Output() close = new EventEmitter<void>();**

**constructor() {}**

**ngOnInit(): void {}**

**onClose(){**

**this.close.emit();}}**

**2-we will create directive which used to host component inside container reference as below**

**import { Directive, ViewContainerRef } from '@angular/core';**

**@Directive({**

**selector: '[appPlaceholder]'})**

**export class PlaceholderDirective {**

**//we inject the view container ref to used to replace component on the target dom element that used directive**

**constructor(public viewContainerRef:ViewContainerRef){}}**

**3-on the auth.component.html we will using ng-template instead of div in order to use template to host container reference**

**<ng-template appPlaceholder></ng-template>**

**//On code behind**

**//used to close subscription for the dyanmic component event emitter**

**//we unsubsribe in 2 placees onDestroy when navigate outside the authcomponent, after click close and remove the dyanmic component**

**closeSub: Subscription;**

**ngOnDestroy(): void {this.closeSub.unsubscribe();}**

**//we will inject component factory resolver to create component at runtime as below**

**constructor(private authService: AuthService,private router: Router,**

**private cmpFactory: ComponentFactoryResolver) {}**

**//it will get the first element that use the directive appPlaceHolder**

**@ViewChild(PlaceholderDirective, { static: false }) alertHost: PlaceholderDirective;**

**//we dynamically create alert component**

**private showErrorAlert(errMsg: string) {**

**const alertCmp = this.cmpFactory.resolveComponentFactory(AlertComponent);**

**//angular needs to container ref : is an object managed by angualr which you set reference place to replace the component there**

**//it will access to view container ref**

**const hostViewContainerRef = this.alertHost.viewContainerRef;**

**//we clear any previous component inside the container ref**

**hostViewContainerRef.clear();**

**//we will attach the new alert component to the container ref (assign component to the container reference as below)**

**const compRef = hostViewContainerRef.createComponent(alertCmp);**

**//to pass values into the created component**

**compRef.instance.message = errMsg;**

**//we have to unsubscribe the subscription becouse the component is then removed**

**//from DOM element , then we clear the container reference from instances**

**this.closeSub = compRef.instance.close.subscribe((param) => {**

**this.closeSub.unsubscribe();**

**hostViewContainerRef.clear();});}**