Structural directives ->zapochvat s \* pr \*ngIf=”courses.length > 0” ili \*ngIf=”DoWehaveCourses()”

<ng-template #noCourses>no courses yet</ng-template>

//no-courses e template var

I sega moje da se kaje

<div \*ngIf="courses.length>0; else noCourses" >list</div>

Hidden – demek scriva go na bazata na uslovie; ama za po-golemi aps e po-dobre da se polzva if, shtoto hidden samo krie

<div [hidden]="courses.length == 0">neshto si</div>

Switch case

Definirame si niakakvo property v clasa

viewMode = 'map';

<div [ngSwitch]="viewMode"> //bindvame go s property ot classa ni

//ako stoynostta mu e map

<div \*ngSwitchCase="'map'">map view</div>

<div \*ngSwitchCase="'list'">list view</div>

<div \*ngSwitchDefault>default view</div>

</div>

**ngFor**

v nego ima t nar exported values pr index; first, last, even, odd

pr even as isEven

<ul>

//slagame valuto na indexa v local var i

<li \*ngFor="let course of courses; index as i">{{course.name}}</li>

</ul>

Pr s even

<ul>

<li \*ngFor="let course of courses; even as even">

{{course.name}}

<span \*ngIf="isEven">{{even}}</span>

</li>

</ul>

TrackBy – mojem da instruktirame angular da polzva razlichen mehanizum za da sledi objects. Vmesto da gi trackva po reference in the memory(koeto mu e defaultnoto povedenie), moje da gi trackva po id) I taka niama da triabva da rerendva tezi elementi povtorno. Tova e mnogo polezno ako imame goliam list I performance problems.

<ul>

//setvame po koj metod da gi trakva, ama ne go callvame ()

<li \*ngFor="let course of courses;trackBy: trackCourse">

{{course.name}}

</li>

</ul>

I posle v app comonenta

trackCourse(index, course){

return course ? course.id : undefined;

}

**ngClass;** vutre pishem key-value pairs, kato vseki key predstavliava css class, a valueto opredelia dali tozi key triabva da bude rendernet ili ne

//valuto e isSelected

<span [ngClass]="{

'glyphicon-star': isSelected,

'glyphicon-star-empty': !isSelected

}">

dff

</span>

**Safe traversal operator – ?** ako assignee e null niama da go pokaje

{{task.assignee?.name}}

**Custom directives -pr ako iskame da () pred vseki nomer**

Ng g d imetoNaDirektivata //g generate, d=directive

//2 posle kat otvorim modules v deklarations otkrivame novata direktiva

//v input.ts

import { Directive, HostListener, ElementRef } from '@angular/core';

@Directive({

selector: '[appInputFormat]' //pri directive imame tia []

})

export class InputFormatDirective {

//we need reference to the host element, thats why we inject element reference object

//tva e service definirana v angular, koiato ni dava dostup do dom objects

constructor(private el: ElementRef) { }

//davame mu imeto na dom eventa

@HostListener('focus') onFocus(){

console.log('focus');

}

@HostListener('blur') onBlur(){

//cheteme valuto na input fielda

let value:string = this.el.nativeElement.value;

this.el.nativeElement.value = value.toLowerCase;

}

}

I v app.component.html

//taka angular shte priloji custom directivata ni

<input type="text" appInputFormat >

A ako iskame I format da polzvame

<input type="text" appInputFormat [format]="'uppercase'">

I sega definirame field v direktivata ni format I go dekorirame s input decoratora

export class InputFormatDirective {

@Input('format') format;

constructor(private el: ElementRef) {}

@HostListener('blur') onBlur() {

let value: string = this.el.nativeElement.value;

if (this.format == 'lowercase') {

this.el.nativeElement.value = value.toLowerCase;

} else {

this.el.nativeElement.value = value.toUpperCase;

}

}

}