# ANGULAR2 TUTORIAL

Angular1 was released in October 2010

Angular2 is Most powerful javascriptis completely rewriten

Angular performance

Performance-Angular performance 5-time faster mobile support testability, component based development

Mobile Support-: With Angular2, we can build a single application that works across mobile and desktop devices

Component based development: In Angular2, everything is component components are the building blocks of an Angular application

More languages choices

1-ECMASCRIPT5

2- ECMASCRIPT6 (Also called E52015)

3-Typescript

4-Dart

5-pure script

6-ETM-etc

Angular2 itself, is built using Typescript: Typescript has great support ECMAScript6 standard

What is ECMASCRIPT?

The JavaScript language standard is officially called ECMAScript

ECMAScript 1 till ECMAScript 7 were released over the past several years

Most modern browsers available today ECMAScript5

The browser support for ECMAScript6 is still incomplete

Trabspilitaion compiles ECMAScript6 to ECMAScript5

ECMAScript6 officially known as ECMAScript 2015

New Features in ECMAScript 2015-Classes, Modules, Arrow Function etc..

What is TypeScript

1 free and open –source programming language developed by Microsoft

2-Superset of JavaScript

3-Transpilation Compiles Typescript to JavaScript

Type script Benefits

1-Intellesence

2-Autocompletion

3-Code navigation

4-Advance refactoring

5-Strong Typing

6-Support ES2015 (ES6) FEATURE like classes, interfaces and inheritance

Severel code editor supports typescript

1-Visual studio

2-Visual studio code

3-Eclipse

4-WebSttorm

5-Atom

6-sublime text etc..

Setting up Angular2 in visual studio

Step 1-Install Node.js and npm

2-Node version 4.6 x or greator

3-npm3.x.x or greator

To Check the version of mode install: node-v

To Check the version of npm installed:npm-v

To Get the latest version of Node.js

<https://nodejs.org/en/download/>

open url after that

After that check system information details

Type window run

Type msinfo32

Check operating system window

Download and install

Step 2-install Visual Studio 2015 Update3

Or Visual studio 2022

Step3 Configure environment setting for node and npm

Step 4-Install Typescript For Visual Studio 2022

To develop, Angular application Typescript

Step 3 Setting Visual studio 2022 in Tools Menu

Go To Option Web Package Management

External web Tools-click

$PATH

Click Arrow Up

Step 4 Install Typescript For Visual studio 2022

To Develop Angular application Typescript 2,2,2 or later is required

2-To Check the version of Typescript, click on help menu vs 2022

3-To Get the latest version of Typescript for Visual Studio 2022

Now Open Visual studio 2022 Go To Help Menu-About Visual studio 2022

Check Typescript

And Create new projects c#-Asp.net wee(net framework)

Angular2demo

Step 6-Download-Quickstart files from Angular web site

Htpp://github.com/angular/quickstart

After that copy and past few important folder

After that run

Cmd-

Cd path write npm start-enter

After that change the

Go TO Src-app-app\_component.ts file open

import { Component } from '@angular/core';

@Component ({

selector: 'my-app',

template: `<h1>Hello {{name}}</h1>`,

})

export class AppComponent { name = 'Angular '; }

After that change name=Angular-to Angular2

After that press cutl+f5 run then after that error is coming

Go to development tool

Check

Step 1-Change index.html- file

<base href=”/”>to point to /src/ folder

<base href=”/src/”>

After that reload page and check error has gone after that show another error

Then change few code index.cshtml page

<script src="/node\_modules/core-js/client/shim.min.js"></script>

<script src="/node\_modules/zone.js/dist/zone.js"></script>

<script src="/node\_modules/systemjs/dist/system.src.js"></script>

After that chane systemjs.config.js

'npm:': '/node\_modules/'

After that Go to tsconfig.ts file and change

What is component in Angular2?

A component in Angular is class with a template and decorator

1-Template

2-Class

3-Decorator

Template defines the user interface . Contains the HTML, Directive and data bindings

Class containyhe code required for the template

Decorator adds meta data to the class making it an Angular Componenet

Step 1 Create class

Export Class AppComponent{

name:string=”Angular”

}

Step 2-Create Decorator syntax

Import {Component} from “@angular/core”

Step 3 Template create syntax

@Component({

Selector:’My-APP’,

Template:`<h1>Hello {{ name }}</h1>`

})

Arrange sequence wise

//First implement class

// second create decorator

import { Component } from "@angular/core"

@Component({

selector: 'my-app',

template:`<h1>hello {{name}}</h1>`

})

export class AppComponent {

name:string="Angular"

}

After that program

See out put

Hello Anvular

//Part5 template vs templateurl

In this session decorator template and templateurl properties

//First implement class

// second create decorator

import { Component } from "@angular/core"

@Component({

selector: 'my-app',

//template: `<h1>hello {{name}}</h1>`//

//or

//template: '<h1>hello {{name}}</h1>'

//or

template: `<h1> single line and double batic character is working single character is not working

hello {{ name }}

</h1>`

})

export class AppComponent {

name: string = "Angular"

Step Now html page inside app folder

App.component.html

}

After that Add new HTML PAGE INSIDE app folder

App.component.html

Cut from code app.componenet.ts template code and past Html page

<h1><span

>hello **{{** name **}}**</span>

</h1>

After that call the html page app.componenet.ts page

Template:’app/app.component.html’

After that run will come error

Now change the code template to templateurl

Then run press cutr f5

And see result

//First implement class

// second create decorator

import { Component } from "@angular/core"

@Component({

selector: 'my-app',

//template: `<h1>hello {{name}}</h1>`//

//or

//template: '<h1>hello {{name}}</h1>'

//or

templateUrl: 'app/app.component.html'

})

export class AppComponent {

name: string = "Angular"

}

Part 6 Angular nested components

In this part Nesting angular component

|  |  |
| --- | --- |
|  | Employee details |
| First name | Tom |
| Lastname | Marggg |
| Gender | Male |
| Age | 29 |

Step 1 Create New Folder Employee-After that right click create Employee.component.html

<table>

<tr>

<td>First Name</td>

<td>**{{**firstname**}}**</td>

</tr>

<tr>

<td>last Name</td>

<td>**{{**lastname**}}**</td>

</tr>

<tr>

<td>Gender</td>

<td>**{{**gender**}}**</td>

</tr>

<tr>

<td>Age</td>

<td>**{{**age**}}**</td>

</tr>

</table>

After that create new Employee-right click-TypeScript-employee.component.ts

import { Component } from "@angular/core"

@Component({

selector: '<my-employee>',

templateUrl: 'app/Employee/employee.componenet.html'

})

export class EmployeeComponent {

firstname: string = 'John';

lastname: string = 'daryy';

gender: string = 'Male';

age: number = 50;

}

Step 2-after that go to app.component.ts file

//First implement class

// second create decorator

import { Component } from "@angular/core"

@Component ({

Selector: 'my-app',

//template: `<h1>hello {{name}}</h1>`//

//or

//template: '<h1>hello {{name}}</h1>'

//or

//templateUrl: 'app/app.component.html'

//template:'<h1> hello {{Headerpage}}</h1>'

template: `

<div>

<h1>{{ pageheader }}</h1>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

pageheader: string = "Employee Details"

}

Step 3-configure the app.module.ts

import { EmployeeComponent } from './Employee/employee.component';

declarations: [ AppComponent,EmployeeComponent ],

now run cunt+f5 and see the result

|  |  |
| --- | --- |
|  | Employee details |
| First name | Tom |
| Lastname | Marggg |
| Gender | Male |
| Age | 29 |

//Part 7 Styling Angular2 Components

This is part seven of Angular2

In this session we will learn

Different options available to apply styles to Angular Componenet

Option in style components

1-Styles in external style sheet:style.css is already link index.html

h1 {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: 250%;

}

table {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

border-collapse: collapse;

}

td {

border: 1px solid black;

padding: 5px 10px;

}

2-stylyes inline in the component HTML file

Derectly apply inside html page

<table style="color: #369; font-family: Arial, Helvetica, sans-serif; font-size: large; border-collapse: collapse;"

>

<tr>

<td style="border: 1px solid black; padding: 5px 10px;">First Name</td>

<td>**{{**firstname**}}**</td>

</tr>

<tr>

<td style="border: 1px solid black; padding: 5px 10px;">last Name</td>

<td>**{{**lastname**}}**</td>

</tr>

<tr>

<td style="border: 1px solid black; padding: 5px 10px;" >Gender</td>

<td>**{{**gender**}}**</td>

</tr>

<tr>

<td style="border: 1px solid black; padding: 5px 10px;">Age</td>

<td>**{{**age**}}**</td>

</tr>

</table>

3-Styles In the component html file using <style>tag

<style>

table {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

border-collapse: collapse;

}

td {

border: 1px solid black;

padding: 5px 10px;

}

</style>

<table>

<tr>

<td>First Name</td>

<td>**{{**firstname**}}**</td>

</tr>

<tr>

<td >last Name</td>

<td>**{{**lastname**}}**</td>

</tr>

<tr>

<td >Gender</td>

<td>**{{**gender**}}**</td>

</tr>

<tr>

<td >Age</td>

<td>**{{**age**}}**</td>

</tr>

</table>

Step 4 Specyfy the styles using the @Component decorator styles property

First comment all style sheet both and direct component based apply

employee.componenet.html

@Component({

selector: '<my-employee>',

templateUrl: 'app/Employee/employee.componenet.html',

Styles: ['table {color: #369;font- family: Arial, Helvetica, sans - serif;font - size: large;border - collapse: collapse;},td {border: 1px solid black;padding: 5px 10px;}']

})

Step 5 Specify the styles using the @component decorator styleUrl property

Step 1 right click Employee folder-add new stylesheet

employee.component.css

table {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

border-collapse: collapse;

}

td {

border: 1px solid black;

padding: 5px 10px;

}

After that employee.component.ts

@Component({

selector: '<my-employee>',

templateUrl: 'app/Employee/employee.componenet.html',

// styles: ['table {color: #369;font- family: Arial, Helvetica, sans - serif;font - size: large;border - collapse: collapse;},td {border: 1px solid black;padding: 5px 10px;}']

styleUrls: ['app/Employee/employee.component.css']// Add that line code

})

Now run program cuntr+f5

//Part8 Interpolation in Angular

In this session interpolation

Interpolation is process Databinding

There is three type

|  |  |
| --- | --- |
| Data Binding | Description |
| One way Data binding | From component to View component |
| One way data binding | From view Template to Component |
| Two way data binding | From component to view & from View template to component |
|  |  |

Step 1 concnating hardcoded string value

App.component.ts

template: `

<div>

<h1>{{'Page Header='+ pageheader }}</h1>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

Now run program cunt+f5

**Page Header=Employee Details**

|  |  |
| --- | --- |
| First Name | John |
| last Name | daryy |
| Gender | Male |
| Age | 50 |
|  |  |

Step 2- perform yhe calculation interpolation

template: `

<div>

<h1>{{10+20+30+40}}</h1>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

# 100

|  |  |
| --- | --- |
| First Name | John |
| last Name | Daryy |
| Gender | Male |
| Age | 50 |

Step 3-ternaary operator

If value is not then no show or value is there then show

template: `

<div>

<h1>{{pageheader?pageheader: ‘No Header'}}</h1>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

pageheader: string = "Employee Details"

value there then show

Employee Details

template: `

<div>

<h1>{{pageheader?pageheader: ‘No Header'}}</h1>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

//pageheader: string = "Employee Details"

pageheader: string = null

}

Value is null then not showing

# No Header

# STEP 4-now show images

template: `

<div>

<h1>{{pageheader?pageheader:'No Header'}}</h1>

<img src={{Imagepath}}>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

//pageheader: string = "Employee Details"

pageheader: string = null

Imagepath: string = 'https://www.pragimtech.com/wp-content/uploads/2019/02/pargim-logo-1.png';

# }

# Step 3 call method function concating string

template: `

<div>

<!--/\*///\*<h1>{{pageheader?pageheader:'No Header'}}</h1>\*///\*/-->

<!--<h1>{{pageheader}}</h1>-->

<h1>{{getfullname()}}</h1>

<img src={{Imagepath}}>

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

pageheader: string = "Employee Details"

//pageheader: string = null

Imagepath: string = 'https://www.pragimtech.com/wp-content/uploads/2019/02/pargim-logo-1.png';

firstname: string = 'Tom';

lastname: string = 'hopkin';

getfullname(): string {

return this.firstname + ' ' + this.lastname;

}

# }

# //Part 9 we will learn property binding Angular2

# Propert binding nd interpolation both data transfer Component to html

# pRoert binding example

# <img [src]='Imagepath'/> propert binding syntax

# Interpolation binding

<img src={{Imagepath}}>

Interpolation v/s Property Binding

# Interpolation is a special syntax that angular converts into property binding

# To Concatanation string we must use interpolation instead of property binding

# I mage path cocatnation

Imagepath: string = 'pargim-logo-1.png';

<img [src]="'https://www.pragimtech.com/wp-content/uploads/2019/02/' + Imagepath" />

template: `

<div>

<!--/\*///\*<h1>{{pageheader?pageheader:'No Header'}}</h1>\*///\*/-->

<!--<h1>{{pageheader}}</h1>-->

<h1>{{getfullname()}}</h1>

<!--<img src={{Imagepath}}>-->

<!--<img [src]='Imagepath'/>-->

<img [src]="'https://www.pragimtech.com/wp-content/uploads/2019/02/' + Imagepath" />

<!-- nested Employee component -->

<my-employee></my-employee>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

pageheader: string = "Employee Details"

//pageheader: string = null

Imagepath: string = 'pargim-logo-1.png';

firstname: string = 'Tom';

lastname: string = 'Hopkin';

getfullname(): string {

return this.firstname + ' ' + this.lastname;

}

# }

# Innerhtml pages using call pageheader

template: `

<div>

<!--/\*///\*<h1>{{pageheader?pageheader:'No Header'}}</h1>\*///\*/-->

<!--<h1>{{pageheader}}</h1>-->

<h1>{{getfullname()}}</h1>

<!--<img src={{Imagepath}}>-->

<!--<img [src]='Imagepath'/>-->

<img [src]="'https://www.pragimtech.com/wp-content/uploads/2019/02/' + Imagepath" />

<!-- nested Employee component -->

<my-employee></my-employee>

<button [disabled]='IsDisable'>Click me</button>

<span [innerHTML]="pageheader"></span>

</div>

`

})

export class AppComponent {

//name: string = "Angular"

pageheader: string = "Employee Details"

//pageheader: string = null

Imagepath: string = 'pargim-logo-1.png';

IsDisable: boolean = false;

firstname: string = 'Tom';

lastname: string = 'Hopkin';

getfullname(): string {

return this.firstname + ' ' + this.lastname;

}

# }

# Part 10 Diffrence between HTML Attribute and DOM Property

# DOM stand for Document object Model when a browser loads a web page,the browser creates a Document object Model of the page

# HTML Element Attribute v/s DOM Property

# 1-Attributes are defined by HTML.Where as properties are defined by the DOM

# 2-Attribute Initialize DOM properties Once the initialization complete ,the attributes jonb is done

# Property values can change where as attribute value cant

# App.component.ts

template: `

<div>

<input id='inputid' type='text' value='Tom'>

# </div>`

# After that run program

# Inspect

# Console

# Inputid.getAttribute(‘value’)

# Tom

# //PART 11 ANGULAR Attribute Binding//

|  |  |
| --- | --- |
| Employee Details |  |
| First Name | John |
| last Name | daryy |
| Gender | Male |
| Age | 50 |

Step 1 Employee folder Create employee1.component.html

<table>

<thead>

<tr>

<th colspan="2">

Employee Details

</th>

</tr>

</thead>

<tbody>

<tr>

<td>First Name</td>

<td>**{{**firstname**}}**</td>

</tr>

<tr>

<td>Last Name</td>

<td>**{{**lastname**}}**</td>

</tr>

<tr>

<td>Gender</td>

<td>**{{**gender**}}**</td>

</tr>

<tr>

<td>Age</td>

<td>**{{**age**}}**</td>

</tr>

</tbody>

</table>

Step 2-Create employee1.component.ts

import { Component } from "@angular/core"

@Component({

selector: "my-employee1",

templateUrl:'app/Employee/employee1.component.html',

styleUrls: ['app/Employee/employee1.component.css']

})

export class employee1Component {

firstname: string = "Tom";

lastname: string = "Hopkins";

gender: string = "Male";

age: number = 40;

}

Step3-Create CSS file employee1.component.css

table {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

border-collapse: collapse;

}

td {

border: 1px solid black;

padding: 5px 10px;

}

thead{

border:1px solid black;

}

Step 4-ADD the emplooye1

app.module.ts

import { employee1Component } from './Employee/employee1.component';

declarations: [AppComponent, EmployeeComponent, employee1Component ],

Step 5-app.component.ts

Call

<my-employee1></ my-employee1>

After that run program cuntr+f5

Press cuntr+f5

Step 2-Colspan interpolation method pass value html page

employee1.component.ts

export class employee1Component {

colspan: number = 2;// Add code and pass the value html

firstname: string = "Tom";

lastname: string = "Hopkins";

gender: string = "Male";

age: number = 40;

}

Step 2- employee1.component.html

<tr>

<th attr.colspan ="**{{**colspan**}}**">

Employee Details

</th>

</tr>

Or

<tr>

<th [attr.colspan] ="colspan">

Employee Details

</th>

</tr>

After that run press cuntr+f5

See result

Part 12 class Binding in Angular2

In this video CSS class binding in Angular

Step 1 GO TO STYLE.CSS

FILE

COMMENT ALL CODE AND NEW STYLE SHEET

.boldClass{

font-weight:bold;

}

.italicClass {

font-style: italic;

}

.ColorClass {

color:red;

}

Step 2-Go to app.component.ts

<div>

<input id='inputid' type='text' value='Tom'>

</div>

<button [ngClass]="'colorClass extraClass'" [class]='classesToApply'>My button</button>

<br/> <br/>

<button [ngClass]="'colorClass extraClass'" [class.boldClass]='applyboldclass'>My button</button>

<br/>

<br/>

<button class="colorsClass" [ngClass]='addclasses()'>My Button</button>export export class AppComponent {

//name: string = "Angular"

pageheader: string = "Employee Details"

//pageheader: string = null

classesToApply: string = 'italicClass,boldClass';

applyboldclass: boolean = false;

applyitalicclass: boolean = false;

addclasses() {

let classes = {

boldClass: this.applyboldclass,

italicClass: this.applyitalicclass

};

return classes

}

Imagepath: string = 'pargim-logo-1.png';

IsDisable: boolean = true;

badhtml: string = 'hello<script>alert("Hacked")</script> world';

firstname: string = 'Tom';

lastname: string = 'Hopkin';

getfullname(): string {

return this.firstname + ' ' + this.lastname;

}

}

Now Run cuntr f5

//Part13 Styling Binding in Angular2//

In this video discussed style binding with Angular2

Template

{

<button style='color:red'>My Button</button>

Second inline style sheet

<button style="color:red"

[style.font-weight]="isBold ? 'bold' : 'normal'">

My Button

</button>

<br/>

<br/>

<button style="color:red"

[style.font-size.px]="fontSize">

My Button

</button>

<br/>

<br/>

<button style="color:red"

[ngStyle]="addstyles()">

My Button

</button>

Export class Appcomponent

isBold: boolean = false;

fontSize: number = 70;

Multiple style apply

addstyles() {

let styles = {

'font-size.px': this.fontSize,

'font-style': this.isitalic ? ‘italic’: 'normal',

'font-weight': this.isBold ? 'bold’: ‘normal',

};

return styles;

}

//Part 14 in this vedioe we will discussed event binding in Angular

These binding flow data in one direction i.e. from a component class property to an

Html element property

1-interpoletion

2-property binding

3-Attribute binding

4-class binding

5-style binding

Event binding flows data in the opposite direction i.e from an HTML Element to a component

Event binding

<button (click)='onClick()'>Click Me</button>

onClick(): void {

console.log('event Clicked');

}

//Part 15 Two way Data Binding in Angular2

In this session two way data binding

Template

{

<!-- part15 one way data binding-->

Name: <input [value]='onewayname'/>

<br/>

Entered: {{onewayname}}

<!--close part15-->

<br/>

<br/>

<!-- part15 two way data binding-->

<input [value]="twowayname" (input)="twowayname = $event.target.value"/>

<br/>

Entered: {{twowayname}}

<!--close part15-->

<br/>

<br/>

<!-- part15 ngmdule way data binding-->

Name: <input [(ngModel)]="ngwayname"/>

<br/>

You Entered: {{ngwayname}}

<!--close part15-->

<br/>

<br/>

Export class Component

{

// part 15 two way data binding//

onewayname: string = 'Tsom';

twowayname: string = 'Tsom';

ngwayname: string = 'Tsom';

Part 16 –Anugar ngfor directives

In This vedieo NgFor directive

Get data from list

Step 1 Employee folder create

Employeelist.component.ts

import {Component} from '@angular/core'

@Component({

selector: 'list-employee',

templateUrl: 'app/Employee/employeelist.component.html',

styleUrls:['app/Employee/emlpoyeelist.component.css']

})

export class EmployeeListComponent {

employees: any[] = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '19/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '20/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '13/09/2019' }

];

}

Step 2 create employeelist.component.html

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date Of Joining</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let employee of employees">

<td>**{{** employee.code **}}**</td>

<td>**{{** employee.name **}}**</td>

<td>**{{** employee.gender **}}**</td>

<td>**{{** employee.annualsalary **}}**</td>

<td>**{{** employee.dateofjoining **}}**</td>

</tr>

</tbody>

</table>

STEP 3 Create css file inside employeelist.component.css

table {

color: #369;

font-family: Arial, Helvetica, sans-serif;

font-size: large;

border-collapse: collapse;

}

td {

border: 1px solid black;

padding: 5px 10px;

}

thead {

border: 1px solid black;

}

Step 4 configure component inside of app.module.ts

import { EmployeeListComponent } from './Employee/employeelylist.component';

declarations: [AppComponent, EmployeeComponent, employee1Component, EmployeeListComponent],

Step 5 call app.cmponent.ts

<list-employee></list-employee>

Now run and see result

Part17 why use TrackBy and ngFor directive

1-How can get the index of an item in a collection

2-identifying the first and last element in a collection

3-identifying even and odd element in a collection

4-why and use TrackNy

1-ngFor directive may perform pooryy with large list

2-Asmal chabge to the list may trigger a xascade of DOM manipulation

Employeelist.component.ts

Step 1

export class EmployeeListComponent {

employess: any[];

constructor() {

this.employess = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '19/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '20/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '13/09/2019' }

];

}

getemployess (): void {

this.employess = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '19/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '20/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '13/09/2019' },

{ code: 'emp105', name: 'Mary', gender: 'Female', annualsalary: 6800.826, dateofjoining: '13/10/2019' }

];

}

}

Step2-employeelist.component.html

<button (click)="getemployess()">Refresh Employess</button>

export class EmployeeListComponent {

employess: any[];

constructor() {

this.employess = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '19/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '20/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '13/09/2019' }

];

}

getemployess(): void {

this.employess = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '19/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '20/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '13/09/2019' },

{ code: 'emp105', name: 'Mary', gender: 'Female', annualsalary: 6800.826, dateofjoining: '13/10/2019' }

];

}

trackByempcode(index: number, employee: any): string {

return employee.code;

}

}

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date Of Joining</th>

<th>Index</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let employee of employess;trackBy: trackByempcode; let i=index i ">

<td>**{{** employee.code **}}**</td>

<td>**{{** employee.name **}}**</td>

<td>**{{** employee.gender **}}**</td>

<td>**{{** employee.annualsalary **}}**</td>

<td>**{{** employee.dateofjoining **}}**</td>

<td>**{{** i **}}**</td>

</tr>

<tr \*ngIf="!employees || employees.length == 0">

<td colspan="5">No Employee data in List</td>

</tr>

</tbody>

</table>

<br />

<button (click)="getemployess()">Refresh Employees</button>

Filter the value even or odd and index filter values

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date Of Joining</th>

<th>Index</th>

<th>Is First</th>

<th>Is Last</th>

<th>Is Even</th>

<th>Is odd</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let employee of employess;trackBy: trackByempcode;index as i;first as isFirst;last as isLast;even as isEven;odd as isOdd">

<!-- Your table cells here -->

<td>**{{** employee.code **}}**</td>

<td>**{{** employee.name **}}**</td>

<td>**{{** employee.gender **}}**</td>

<td>**{{** employee.annualsalary **}}**</td>

<td>**{{** employee.dateofjoining **}}**</td>

<td>**{{** i **}}**</td>

<td>**{{** isFirst **}}**</td>

<td>**{{** isLast **}}**</td>

<td>**{{** isEven **}}**</td>

<td>**{{** isOdd **}}**</td>

</tr>

<tr \*ngIf="!employees || employees.length == 0">

<td colspan="5">No Employee data in List</td>

</tr>

</tbody>

</table>

<br />

<button (click)="getemployess()">Refresh Employees</button>

Part 18 in this session we will learn Pipes Angular2

Angular2 is pipe is transform the data lower case to Uppercase and filter the data currency decimal as expected values

First lower case to Uppercase

Pipe in Angular

Transform data before display

Built in pipes include lowercase,uppercase,decimal,date,percent,currency etc

To APPLY A PIPE ON A BOUND PPROPERTY USE THE PIPE CHARACTER “|”

Angular PIPE

Date : [https://angular.io/api/common/DatePipe](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbXp0ZnM3bUFKdUhzQXdXeW5McVl6dVlUd0RVZ3xBQ3Jtc0tuRUJNRGxxYjk1dEJkRU5ldHB2ZEhNWmhiRVlRZ3FxMm5fREE4bU5VV0RMd2dTcldRaUdCSk81WU9SRkliZV9mZXZyZEc2OWtmTVhqeWVsNGdZZDRFaXJfdFJGSldrWGN6NmN3RWhPR1BhLXpYQ0lnSQ&q=https%3A%2F%2Fangular.io%2Fapi%2Fcommon%2FDatePipe&v=jAUwKOLNpjQ)

Decimal : [https://angular.io/api/common/Decimal...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbW9HMDFXWmw0Vlh5bVlEQUJRSDc3czk4QVZIQXxBQ3Jtc0ttZlZmd21UWGRRZVozSWFLUmhYQndkdkpYOW5xY3hmd1M2NGYwbDFFeHVLcmllTmo4bjZIU0hmc3ZtSXhveU9ZekEzRURJcHRoVTFNd3oyMHdHZGNGemRQeTNLcWJvcWx0cGJHUHVEcGU5VjVTdDFlNA&q=https%3A%2F%2Fangular.io%2Fapi%2Fcommon%2FDecimalPipe&v=jAUwKOLNpjQ)

Currency : [https://angular.io/api/common/Currenc...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqazFKYWsxQnJjcjZEOURNcThkYy03dGhaSW5kUXxBQ3Jtc0trdi1Jc0lRYjVBOGlPV0wzVm9aak5GN0VjZTl3YkR6TXBHbFZ1SkZzakoza3hBaUZ0R2NKckg1ZllOdmdzRW5ieEF3X2c5Z0Q5QkZGOWtHM0E5ci1UTDYyVUdSVlVJN3VUaWhhTENxNUNPdnJvOUdMNA&q=https%3A%2F%2Fangular.io%2Fapi%2Fcommon%2FCurrencyPipe&v=jAUwKOLNpjQ)

Percent : [https://angular.io/api/common/Percent...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbHhadF93WWY0Zy1INW1TUEhDNTVjLWMzb0ZuUXxBQ3Jtc0tuVkVVOFh0Y25OWXFnbkxUWGFveHZCeWgzY3V3RU4tVUtNS2l3VnQ5M0d2V0JTVURjM2J4M0hrZXptT1lVNkllVlNWa183RXFfY0gybFNwalJ5cmVBTnZoNktoNDVmaXFockRPOTNWUmhsSzBoejZEaw&q=https%3A%2F%2Fangular.io%2Fapi%2Fcommon%2FPercentPipe&v=jAUwKOLNpjQ)

<tbody>

<tr \*ngFor="let employee of employess;trackBy: trackByempcode;index as i;first as isFirst;last as isLast;even as isEven;odd as isOdd">

<!-- Your table cells here -->

<td>**{{** employee.code | uppercase **}}**</td>

<td>**{{** employee.name |uppercase **}}**</td>

<td>**{{** employee.gender **}}**</td>

<td>**{{** employee.annualsalary | currency:'INR':true:'1.3-3' **}}**</td>

<!--<td>**{{** employee.dateofjoining | date:'fullDate'|uppercase **}}**</td>-->

<td>**{{** employee.dateofjoining | date:'dd/MM/y'|uppercase **}}**</td>

<td>**{{** i **}}**</td>

<td>**{{** isFirst **}}**</td>

<td>**{{** isLast **}}**</td>

<td>**{{** isEven **}}**</td>

<td>**{{** isOdd **}}**</td>

</tr>

// PART 19 CUSTOM PIPES IN Angular 2

//In this vedio create custom pipes//

Fixed pipe pervious gender or male and female

Mr or Ms.

Step 1 right click Employee folder-employeeTitle.pipe.ts

employeeTitle.pipe.ts

import { Pipe, PipeTransform } from "@angular/core";

@Pipe({

name: 'employeetitle'

})

export class EmployeeTitlePipe implements PipeTransform {

transform(value: string, gender: string): string {

if (gender.toLowerCase() == 'male') {

return 'Mr.' + value;

}

else {

return 'Miss.' + value;

}

}

}

Step 2- <td>**{{** employee.name | employeetitle:employee.gender **}}**</td>

Step 3- app.module.ts

import { EmployeeTitlePipe } from './Employee/employeeTitle.pipe';

declarations: [AppComponent, Employee Component, employee1Component, EmployeeListComponent, EmployeeTitlePipe],

run

Part 20 in this vedio Angular Container and nested Component

What is nested component

What is a container component

Passing data from the nested component to container component

Passing data from the container component to nested component

Along the way we will discuss component input and output properties

Creating custom events using EventEmitter class

What is ng-container directives and it is use

Employee-older-Create-new empoloyee.component.ts

import { Component } from '@angular/core'

@Component({

selector: 'employee-count',

templateUrl: 'app/Employee/employeecount.component.html',

styleUrls:['app/Employee/employeecount.component.css']

})

export class EmployeeCountComponent {

all: number = 10;

male: number = 5;

female: number = 5;

}

Step 2 Create css file

employeecount.component.css

.radioClass{

color:#369;

font-family:Arial,Arial, Helvetica, sans-serif;

font-size:large;

}

Step 3-employee.component.html

<span class="radioClass">Show:</span>

<input type="radio" name="options" />

<span class="radioClass">**{{**"All(" +All+")"**}}**</span>

<input type="radio" name="options" />

<span class="radioClass">**{{**"Male("+Male+")"**}}**</span>

<input type="radio" name="options" />

<span class="radioClass">**{{**"Female("+Female+")"**}}**</span>

Part 21 In this session we will learn

How to pass data from the container components to the nested component using input properties

Employeelist.component.ts

import {Component} from '@angular/core'

@Component({

selector: 'list-employee',

templateUrl: 'app/Employee/employeelist.component.html',

styleUrls: ['app/Employee/emlpoyeelist.component.css']

})

export class EmployeeListComponent {

employess: any[];

constructor() {

this.employess = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '02/12/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '02/11/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '02/10/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '02/10/2018' }

];

}

getemployess(): void {

this.employess = [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '11/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '10/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '11/09/2019' },

{ code: 'emp105', name: 'Mary', gender: 'Female', annualsalary: 6800.826, dateofjoining: '10/10/2019' }

];

}

getTotalEmployeeCount(): number {

return this.employess.length;

}

getTotalMaleEmployeeCount(): number {

return this.employess.filter(e => e.gender === "Male").length;

}

getTotalFeMaleEmployeeCount(): number {

return this.employess.filter(e => e.gender === "Female").length;

}

|  |  |
| --- | --- |
| Operator | Use to |
| = | Assign to value |
| == | Compare two values |
| === | Compare two values and their types |

Employeelist.component.ts

getTotalEmployeeCount(): number {

return this.employess.length;

}

getTotalMaleEmployeeCount(): number {

return this.employess. Filter(e => e.gender === "Male").length;

}

getTotalFeMaleEmployeeCount(): number {

return this.employess.filter(e => e.gender === "Female").length;

}

onEmployeeCountRadioButtonChange(selectedRadioButtonValue: string): void {

this.selectedEmployeeCountRadioButton = selectedRadioButtonValue;

}

getFilteredEmployees(): any[] {

if (this.selectedEmployeeCountRadioButton === 'All') {

return this.employess;

} else {

return this.employess.filter(e => e.gender === this.selectedEmployeeCountRadioButton);

}

}

Step2 employeelist.component.html

<employee-count [all]="getTotalEmployeeCount()"

[male]="getTotalMaleEmployeeCount()"

[female]="getTotalFeMaleEmployeeCount ()">

</employee-count>

<br/>

<tr \*ngFor="let employee of getFilteredEmployees();

====================================================================================//Part 23 Interfaces in Angular2

In this tutorial video

What is interface in Angular2?

1-Use interface Anyword to create an interface

2-it is to common to prefix the interface name with capital letter:However some

Interface in Angular does not have the points for example OnInit interface

Interface member by default public and does not return explicit access modifiesr it is compile time error to include in explicit access modifier interface

A class that interface and interface must provide implementation for all the interface member unless the member are marked as optional keyword

Step right click Employee folder-addd new type script file

Employee.ts

After that create interface

export interface IEmployee {

code: string;

name: string;

gender: string;

annualsalary: number;

dateofjoining: string;

/\*department?: string;// optional propert\*/

/\*computemonthlysalry(annualsalary: number): number;\*/

}

export class Employee implements IEmployee {

constructor(public code: string, public name: string, public gender: string

, public annualsalary: number, public dateofjoining: string) {

}

//computemonthlysalry(annualsalary: number): number{

// return annualsalary / 12;

//}

}

AND After that Import interface employeecount.ts

import { IEmployee } from './employee';

employees=IeMPLOYEE[] NOT USING Any keword

After that automatically show the inelegance

//Part 24 Angular component Lifecycle

In this video discuss angular component lifrcycle

Create the component

Renders the component

Create and renders component children

Check when the component data bound properties change and

Destroy the component before therefore removing it from the DOM

Angular offer several lifecycle hooks

Ngonchanges

ngOminit

ngOnCheck

ngonchnages-Execute every time the value of an input property changes the hook method receive the simple changes object containing current and previous

property values.This is called before ngOninit

ng OnINIT—execute offer the constructor and after ngOnChanges hook for the first time it is most commonly used component initialization and retrieving from data database

ngOnDestroy—Execute just before angular destroys the component and generally used the performing change

OnChanges

Step 1-Create A new Folder Others-Right click Add new type script file

Simple.Component.ts

import { Component,Input,OnChanges,SimpleChange, SimpleChanges } from '@angular/core'

@Component({

selector: "simple",

template:`You entered:{{simpleinput}}`

})

export class SimpleComponent implements OnChanges {

@Input() simpleinput: string;

ngOnChanges(changes: SimpleChanges) {

for (let PropertyName in changes) {

let change = changes[PropertyName];

let current = JSON.stringify(change.currentValue);

let previous = JSON.stringify(change.previousValue);

console.log(PropertyName + "+currentValue" + current + "+previousValue+" + previous);

}

}

}

After app.component.ts

Your text:<input type ='text' [(ngModel)]='userText'/>

<br/>

<br/>

<simple [simpleinput]='userText'></simple>

And

'userText':string

Part 25 Angular Services Tutorial

In this vedieo we will learn

1-Why need service

2-Creating Service

3-Injecting and using a service

4-Diffrence between constructor and ngOnInit

Service-A service in Angular is generally used when you need to reuse data or logic across multiple component

Step 1 Create first employee.service.ts

File type script(service is nothing a class)

employee.service.ts

import { Injectable } from '@angular/core';

import { IEmployee } from './employee'// interface injecte here

@Injectable()

export class EmployeeService {

getemployess(): IEmployee[] {

return [

{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '11/02/2015' },

{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '11/06/2016' },

{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '10/06/2017' },

{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '11/09/2019' }

];

}

}

Emploloyeelist.component.ts

import { EmployeeService } from './employee.service';

@Component({

selector: 'list-employee',

templateUrl: 'app/Employee/employeelist.component.html',

styleUrls: ['app/Employee/emlpoyeelist.component.css'],

providers: [EmployeeService]

})

constructor(private \_employeeservice: EmployeeService) {

//this.employess = [

// //{ code: 'emp101', name: 'Tom', gender: 'Male', annualsalary: 5500, dateofjoining: '02/12/2015' },

// //{ code: 'emp102', name: 'Alex', gender: 'Male', annualsalary: 5700.98, dateofjoining: '02/11/2016' },

// //{ code: 'emp103', name: 'Mike', gender: 'Male', annualsalary: 5900, dateofjoining: '02/10/2017' },

// //{ code: 'emp104', name: 'Mary', gender: 'Female', annualsalary: 6500.826, dateofjoining: '02/10/2018' }

//];

// this.employess = this.\_employeeservice.getemployess();

}

ngOnInit() {

this.employess = this.\_employeeservice.getemployess();

}

App.component.ts

<list-employee></list-employee>

After that run

Part 26 Angular and ASP.NET WEB api

Intis vedieo

Createing ASP.NET WEB API SERVICE

Step 1 SATART SQL CREATE TABLE

Create table EmployeeAngularservice

(

code nvarchar(50) primary key,

name nvarchar(50),

gender nvarchar(50),

annualsalary decimal(18,3),

dateofjoining nvarchar(50)

)

insert into Employee values( 'emp101', 'Tom', 'Male', 5500, '02/12/2015')

insert into Employee values('emp102', 'Alex', 'Male', 5700.98, '02/11/2016')

insert into Employee values('emp103', 'Mike', 'Male', 5900, '02/10/2017')

insert into Employee values('emp104', 'Mary', 'Female', 6500.826, '02/10/2018')

SATEP 2-Now add new project right click solution explorer Add new item-Asp.net web application-select-template-Web Api –se

WebApiAngular

Right clclick on webApiAngular-add new projects-DATA-Ado.net entity model-name-EmployeeDataModel—of-next connection string

After that Go to controller add new item web api EmoloyeesController

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

namespace WebAppiangular.Controllers

{

public class EmployessController : ApiController

{

public IEnumerable<EmployeeAngularservice> Get()

{

using (AngularnetcoreEntities2 entities2 = new AngularnetcoreEntities2())

{

return entities2.EmployeeAngularservices.ToList();

}

}

public EmployeeAngularservice Get( string code)

{

using (AngularnetcoreEntities2 entities2 = new AngularnetcoreEntities2())

{

return entities2.EmployeeAngularservices.FirstOrDefault (e=>e.code==code);

}

}

}

}

Cuntr+f5=

Run projects

Part 27 Angular 2 http services Tutorial

In this part 27 of angular 2 tutorial how to call Asp.net web api in using Angular 2

Observable

Call the api server side in using built in Angular

Step 1 within the root module (AppModule), Import the angular HTTP module

Open

app.module.ts

Import {} from HttpModule

Import { HttpModule } from '@angular/http';

and also include

imports: [BrowserModule, FormsModule, HttpModule],

Step2-Modify EmployeeService to issue a GET request using the built –in http service

Step 3-employee.service.ts

Include package

import {Http, Response } from '@angular/http'//Part 27 call web api using angular

import { Observable } from 'rxjs/Observable';//part 27

import 'rxjs/add/operator/map';

After that get data

export class EmployeeService {

constructor(private \_http: Http) { }

getemployess(): Observable<IEmployee[]> {// part 27 call api

return this.\_http.get("http://localhost:52116/api/Employess")

.map((response: Response) => <IEmployee[]>response.json())

}

}

What is an observable

Observer also called (Subscribers) subscribe to

The callback function is notified when the Observable emits data

emits data

Observer

Observer

Observer

Observable

Step 3 How can subscribe of Observable