Angular tutorial

Template syntax :- Architcture :

* Module :- Represents Feature area of applicaton. Every angular application has atleast One module which is root module is called as app module
* Component : - Each module is intern made of component and services Each component will have HTML template which represents the view in the browser and class controllers the logic

of particular view

Module will have service which have business module of your application

1] Angular app ontain one or more module

2] module contain one or more components and services

3] components contain HTML template for view and contain class

for logic for tat particular view

5] Modules interact and ultimetly render view in the browser

File Structure :-

Package.json :- This file contains dev dependencies

Main.ts : - it is Entry point of angular app .

SRC folder :- 1] app Folder :-

* It contains app.module.ts file which is the root module of our angular application
* And also contain app.component.ts file which is root component of angular application

Component In Brief :-

Component made of 3 part

1. Template : - it represents the view It is made of HTML and using this

we are creating the user interface of

1. Classs :- it is nothing but code which supports the view and this is created using typescript It contains data & Methods
2. MetaDAta : - It is nothing but the decorator it is basically function that attaches the class right below it It tells the angular it is not plane template

e.g.

**app.component.ts**

@Component({      ///  This is a Decorator which means it is a Metadata its

  selector: 'app-root',

  // it is custom  Html tag that can represent the current component we can use it as class e.g selector:'.app-root' and simply add class to div in app.component

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

 // It points to html that represents the view of current component and template represents the html tag i.e templateUrl: `<h1>Hello</h1>`

  styleUrls: ['./app.component.css']

   // It points to the css file for current component

})

App.component.html : - It is represents a view for app component(i.e for angular app same as like in ReactJS) angular app contain one root component which is app.component.html and other components will fall under this app component

e.g.

App.component.html

<span>{{ title }} app is running!</span>

<app-first-attempt></app-first-attempt>

InterPolation :- if we want to render name or any other property dynamically the we use interpolation there we use interpolation in test.component.ts and display it on app.component.html

e.g.

**test.component.ts**

@Component({

  selector: 'app-test',

  template: `<h1>Hello {{name}}</h1>`, // {{name}} this property is called as interpoaltion syntax

  styleUrls: ['./test.component.css']

})

export class TestComponent implements OnInit {

  public name = "pradeep" //Here we can declare the name property

  constructor() { }

  ngOnInit() {

  }

}

And app.component.html looks like as follows

e.g.

<span>{{ title }} app is running!</span>

<app-test></app-test>

We can do anything in interpolation please see the below image you may understand that

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

@Component({

  selector: 'app-test',

  template: `<h1>Hello {{name}}</h1>

              {{"welcome " + name}}

              {{2+2}}

              {{name.length}}

              {{greetUser()}} `,// binding the method

              // {{name}} this property is called as interpoaltion syntax here we can do couple of thing like Concatination and mathematical operation

  styleUrls: ['./test.component.css']

})

export class TestComponent implements OnInit {

  public name = "pradeep"

  constructor() { }

  ngOnInit() {

  }

  greetUser=()=>{

    return "Function Hello " + this.name

  }

}

Property Binding :-

Difference betwrn Attribute and Property

* Attributes and property are not same
* Attributes define by the HTML
* Properties Definesby the DOM (document obeject model)
* Attributes initialise Dom properties and then they are done. Attributes values cannot change once they initialised
* Property values however change

Please see the image below for better understandin of property binding

We add id and disabled attribute in square bracket to know the browser or html which operation to be perform because we put disabled=”false” then textbox is in disabled condition but when we use property binding i.e. [disabled]=”false” then it works instead of [] we use bind property

i.e. bind-disabled=”false”

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

@Component({

  selector: 'app-test',

  template: `<h1>Hello {{name}}</h1>

              <input type="text" [id]="myID" value="pradeep">

              <input type="text" [disabled]="false" id={{myID}} value="pradeep">` ,//[id] it is known as property binding

  styleUrls: ['./test.component.css']

})

export class TestComponent implements OnInit {

  public name = "pradeep"

  public myID ="test"

  constructor() { }

  ngOnInit() {

  }

  greetUser=()=>{

    return "Function Hello " + this.name

  }

}

Class Binding :-

In class binding we can add css class to the HTML Element and see the file is below

Test.component.ts

@Component({

  selector: 'app-test',

  template: `<h1 class="text-success">Hello {{name}}</h1>

             ` ,//[id] it is known as property binding

  styles: [`

              .text-success{

                color:green;

              }

          `]

})

Style Binding :- It pretty similar to class binding

Test.component.ts

@Component({

  selector: 'app-test',

  template: `<h1 class="text-success">Hello {{name}}</h1>

              <span [class.text-danger]= "hasError"> Pradeep</span>

              <h2 [ngClass]="specialClass">Pradeep</h2>

              <h2 [style.color]="'orange'">Style Binding</h2>

              `//this is stylebinding

             ,//[id] it is known as property binding

  styles: [`

              .text-success{

                color:green;

              }

              .text-danger{

                color:red;

              }

          `]

})

We can assign conditionally assign the value to style binding see image below

@Component({

  selector: 'app-test',

  template: `<h1 class="text-success">Hello {{name}}</h1>

              <span [class.text-danger]= "hasError"> Pradeep</span>

              <h2 [ngClass]="specialClass">Pradeep</h2>

              <h2 [style.color]="hasError ? 'red' : 'green'" >Style Binding</h2>

              `//this applly class conditionally

             ,//[id] it is known as property binding

  styles: [`

              .text-success{

                color:green;

              }

              .text-danger{

                color:red;

              }

          `]

})

We assign color to style Binding text using turnary operator if hasError is true the color should be red else It should be green

Event Binding : -

In Angular 8, event binding is used to handle the events raised from the DOM like button click, mouse move etc. When the DOM event happens (eg. click, change, keyup), it calls the specified method in the component. Using $event parameter we can get the event property . Event binding captures in DOM event and perform specific action

Please see the image below for better understanding.

e.g.

test.component.ts

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

@Component({

  selector: 'app-test',

  template: `<button (click)="Submit($event)">Click</button> {{greeting}}`,

  styles: [`

              .text-success{

                color:green;

              }

              .text-danger{

                color:red;

              }

          `]

})

export class TestComponent implements OnInit {

  public name = "pradeep"

  public greeting = ""

  constructor() { }

  ngOnInit() {

  }

  greetUser=()=>{

    return "Function Hello " + this.name

  }

  Submit(event){

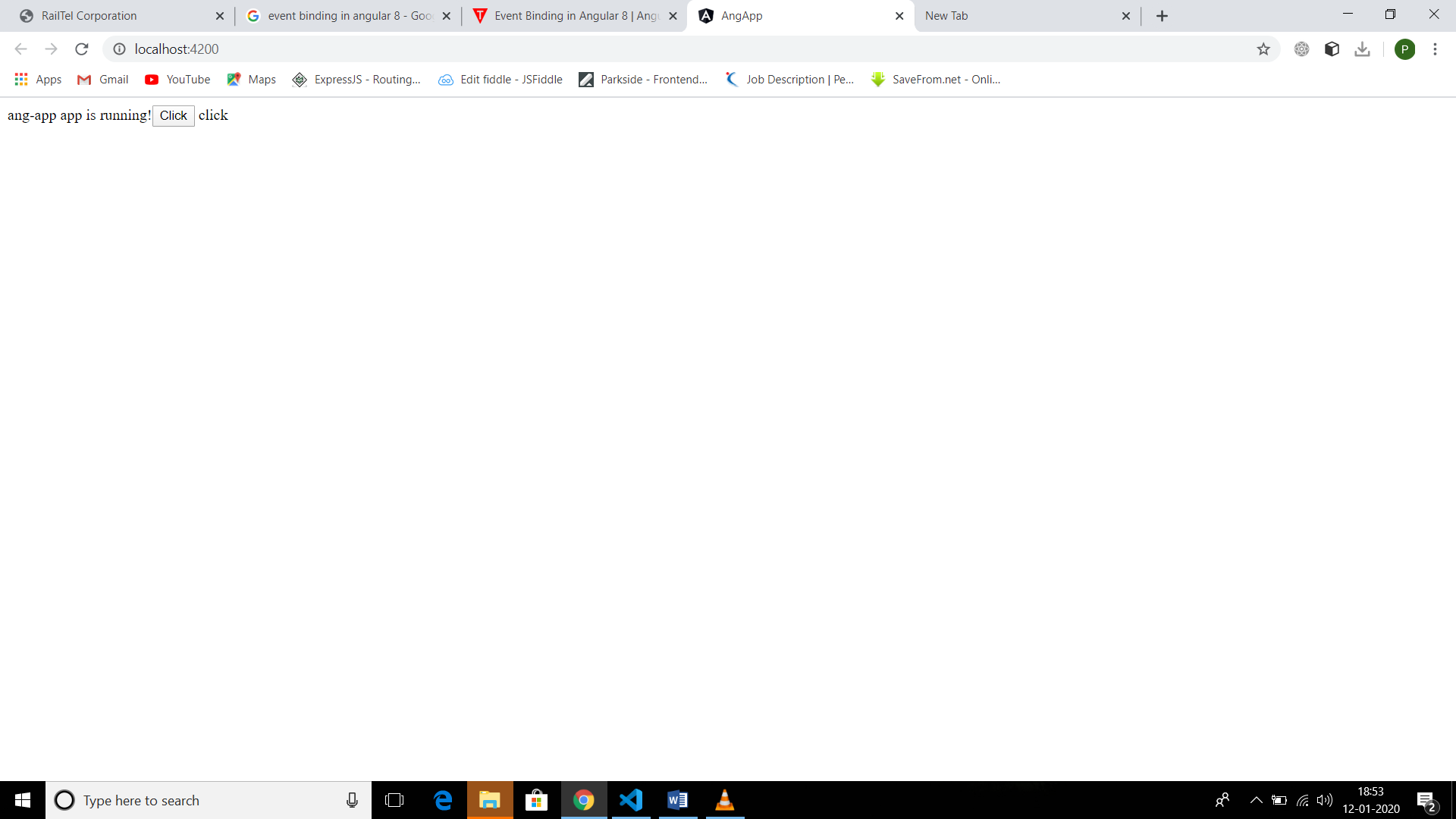
    console.log("You Clicked")

    this.greeting = event.type

  }

}

Using {{greeting}} we show the event property on click see image below



Template Refrence Variable