Day 31 Revisit

1. NodeJS – It’s Open Source JavaScript Runtime Environment
2. NPM – Node Package Manager (Dependency mgmt. tool for JS projects)
3. SPA – Single Page Application (Created With any JS based framework like angular, react, vue etc.,)
4. TypeScript – Typed Super Set of JavaScript (Strongly typed language that is developed by Microsoft)

Day 32 Agenda

1. Angular CLI – Installation, creating & running angular application
2. Components, Directives, Decorators
3. Structure of an angular application
4. Data types, important files & Testing

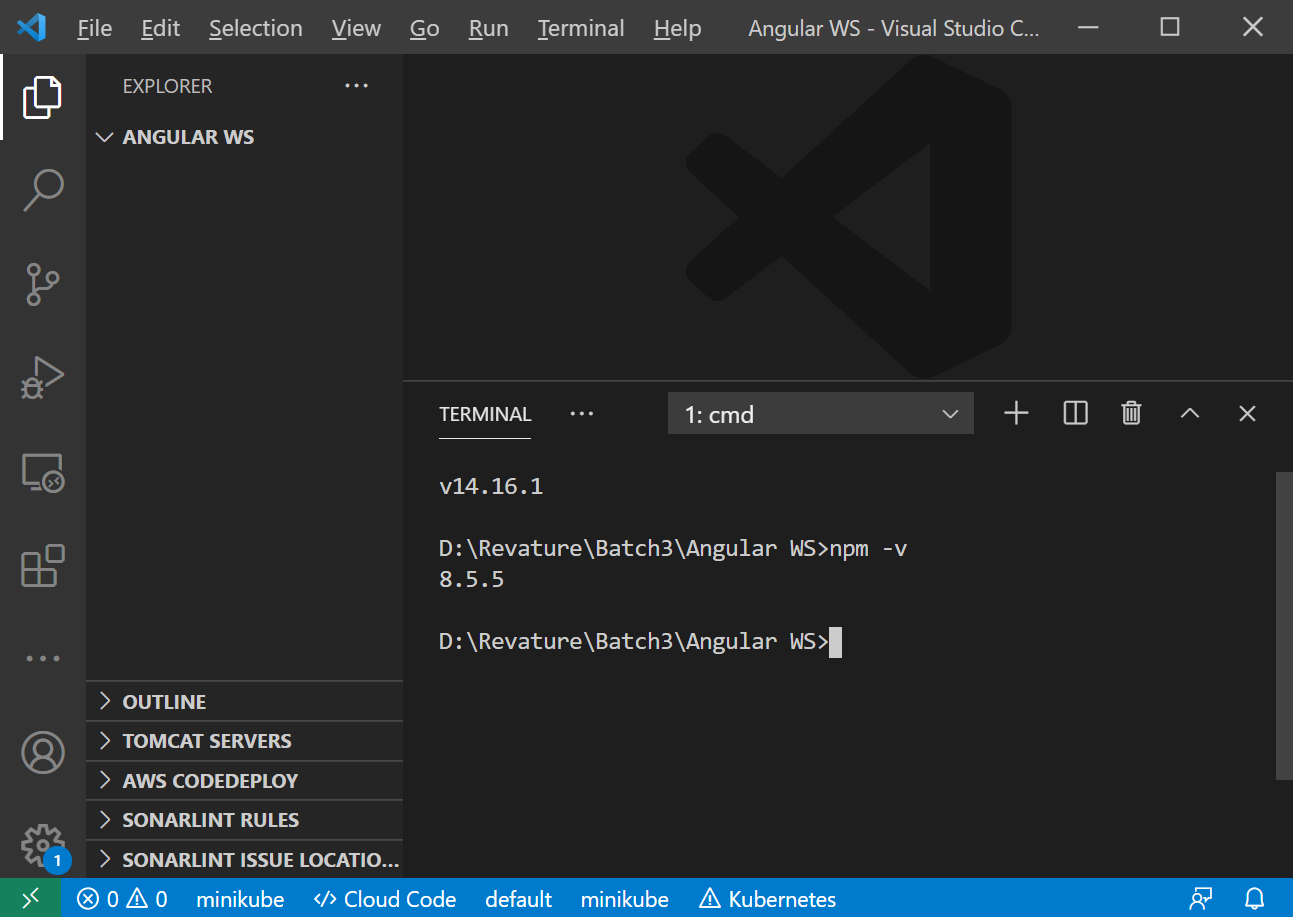
NodeJS & NPM

For all the front end based frameworks the best suited IDE is VS Code.

Angular is a very popular open source, JS based front end framework developed by google.

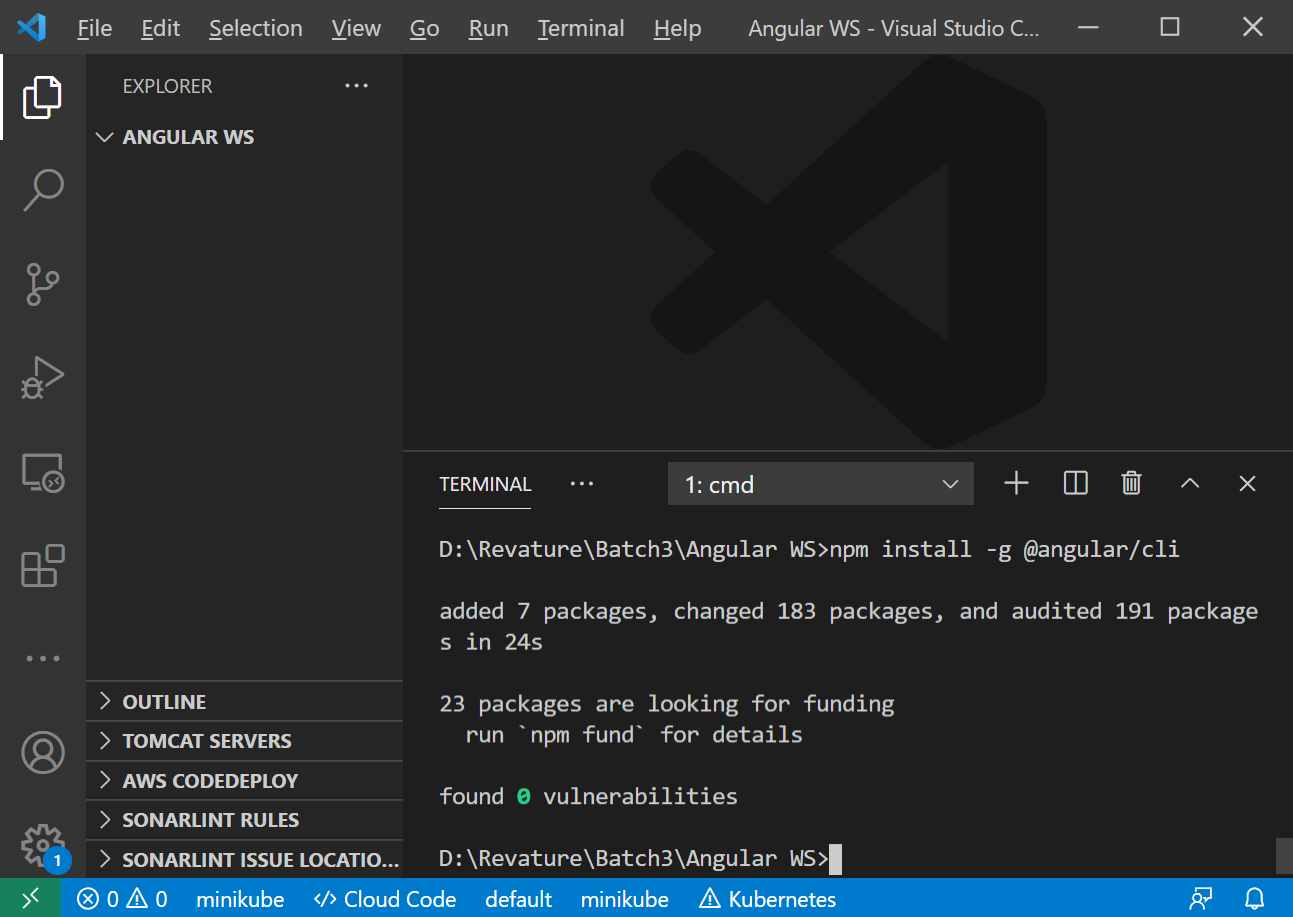
Creating Angular App

Check Node & npm version in VS code

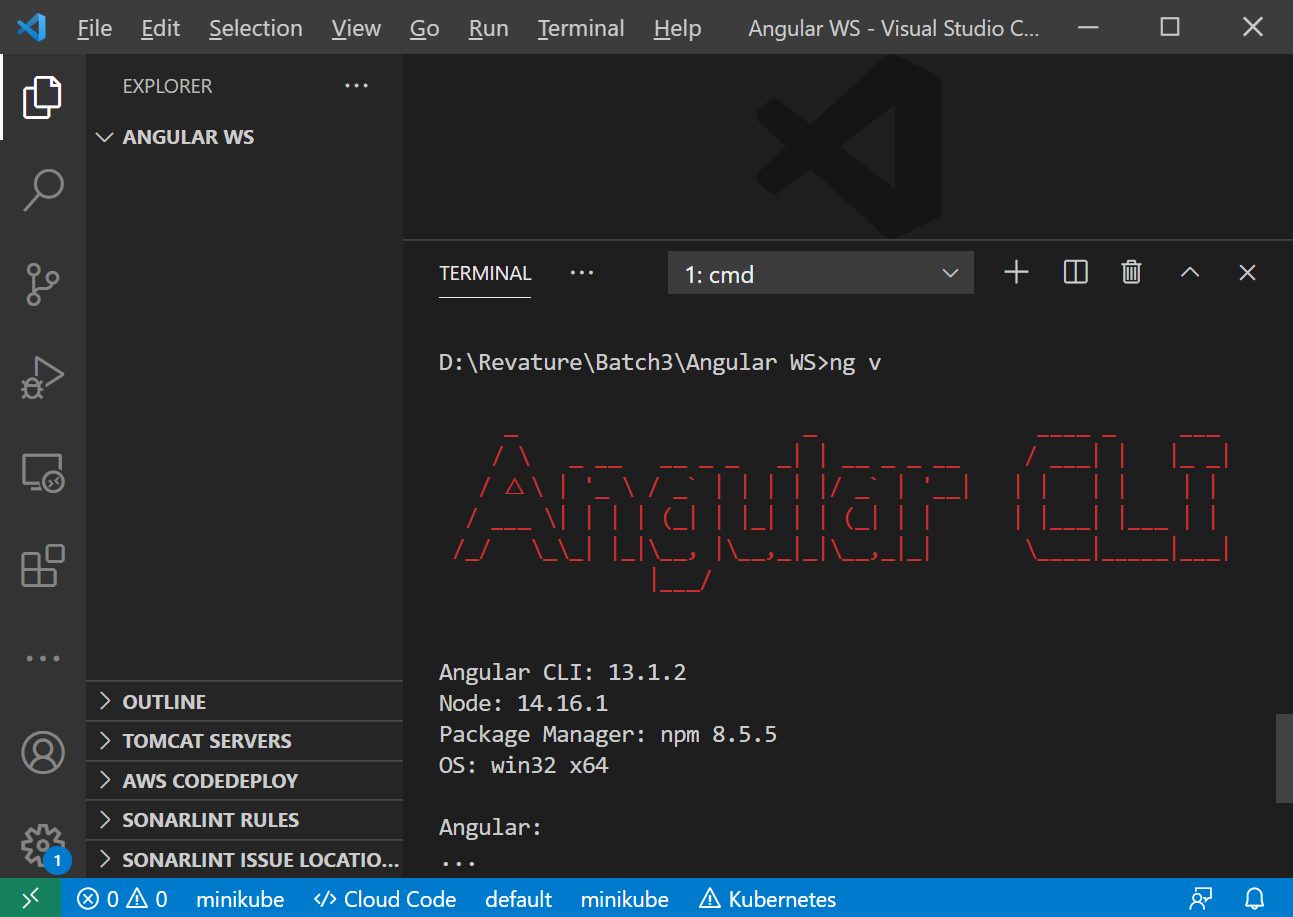


Install Angular CLI ( Command Line Interface)

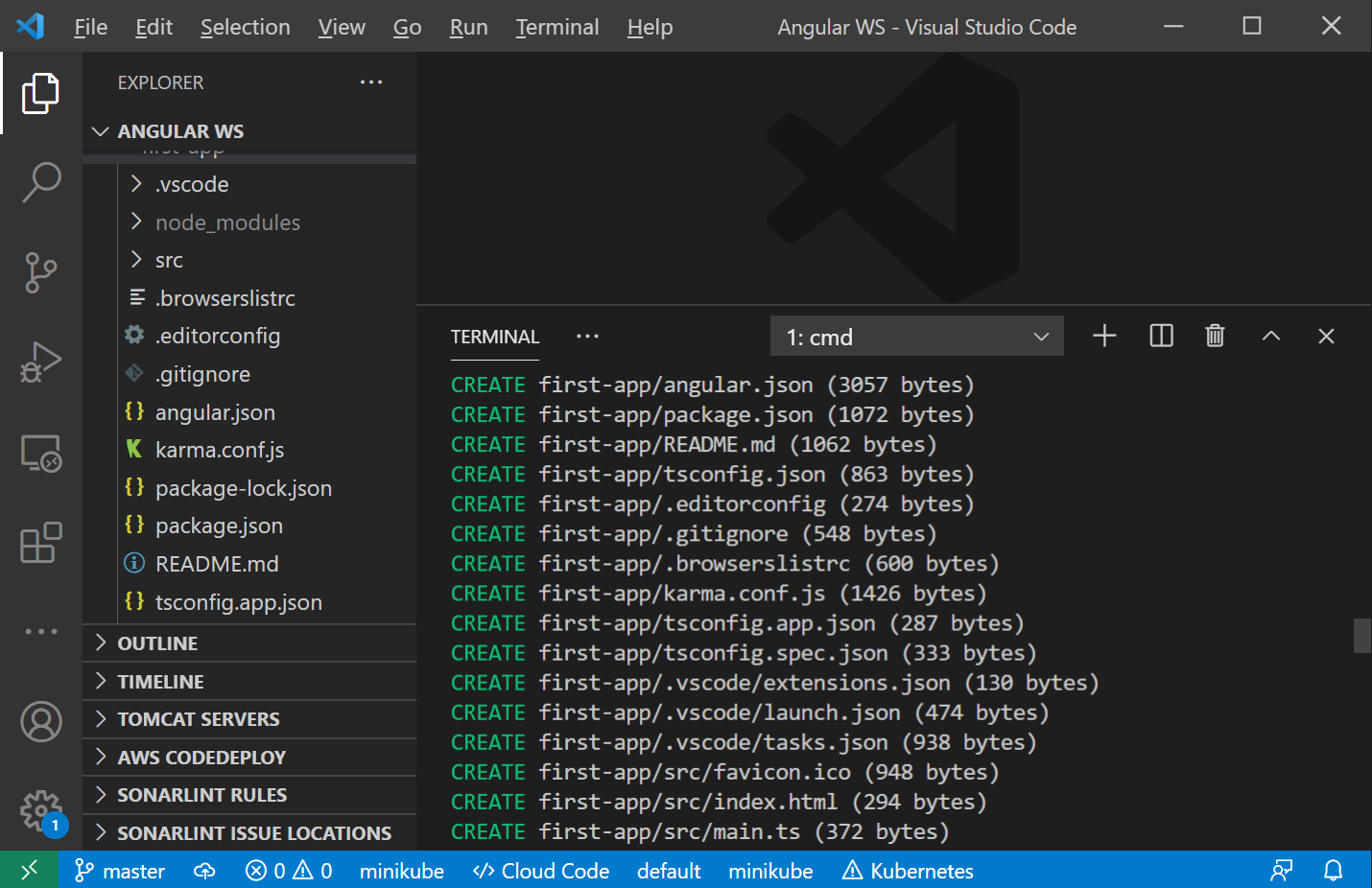
Npm install -g @angular/cli



Ng v



Ng new <app-name>



Src -folder – all the application related codes (Components, routers, services, interfaces)

Node\_modules – all the angular app dependencies will be downloaded locally

Components – Reusuable piece of code (It has both html, css & js/ts)

SPA = Only one A4 Sheet on top of it many sticky note (component)

SPA – Single page application have only one HTML file called index.html (Landing page)

HTML tags are pre-defined

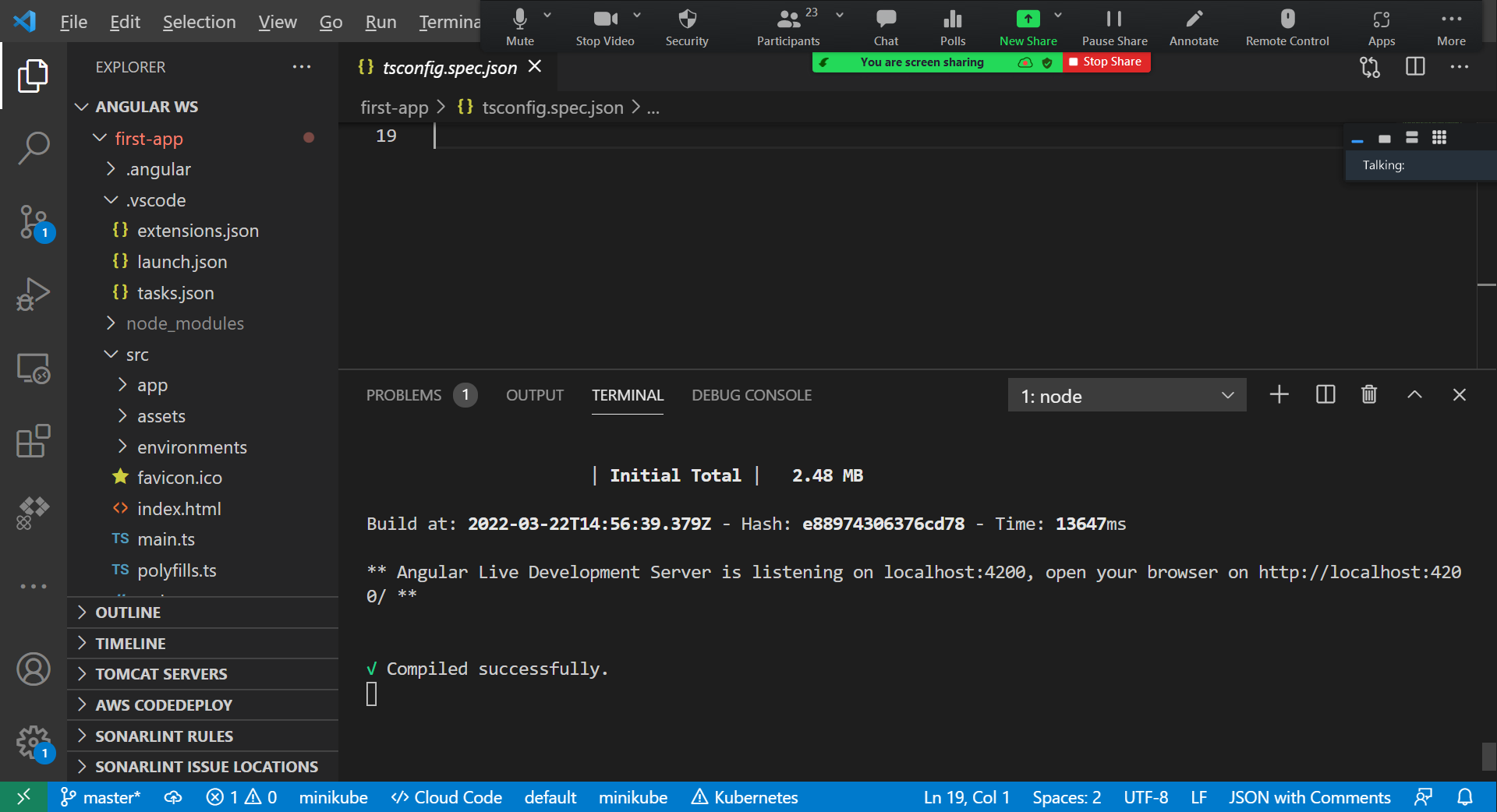
With the help of angular, we can create custom html tags

<siva> </siva>

To run an angular application

Cd first-app

Ng serve -o



Angular Application

Root Module ( app.module.ts) typescript file

Root Component ( app.component.ts) typescript file

Angular uses typescript language for writing component, services, modules and router codes.

Decorator == Similar to annotations in Java (It provides meta data )

@NgModule

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

Each component will have 4 files ( 2 Mandatory files (ts, spec.ts) , 2 optional files (template [html], style [css])

let & const are keyword in typescript

let x:number = 25; // A variable declaration and initialization

let y:string; // A string variable declaration

let isActive = false; //Boolean variable declaration & initialization

TypeScript data types

1. boolean
2. number
3. string
4. undefined & null
5. array
6. tuple
7. any
8. never

Access Modifiers

1. public (Default)
2. private
3. protected
4. readonly

Class – class keyword is used to create class

Class can be extended using extends keyword

Class can have constructors & get and set methods [Accessors & Mutators]

Decorators – Are meta data – similar to annotation in Java

**Types of Decorators**

There are 4 different types of decorators:

1. Class decorators
2. Method decorators
3. Property decorators
4. Parameter decorators

WebPack = It’s a Module Bundler (Tool) – bundling several files to a single file

Angular CLI – Angular Command Line Interface – To manage Angular Application (Create, add, remove, edit, run, test, package the angular application)

Command V CLI (Command will do a single operation like creating a file, copying file content, renaming file, moving a file to different location, creating a folder,

Where as CLI will help in creating the application, running it, testing it – It will do more than a single operation )

Angular App folders

1. E2e – For all End to End Test cases
2. Node\_modules – For all the dependencies
3. Src – For all the application source code

Files

1. Package.json – contains details about the project and all its dependencies

Angular will release a new version in once in 6 months.



Binding – providing data to the view by the model ( Sending data from ts to html file or sending data (events) to ts file)

Component - View (HTML) & Data (Model .ts [class/interface])

Types of Binding

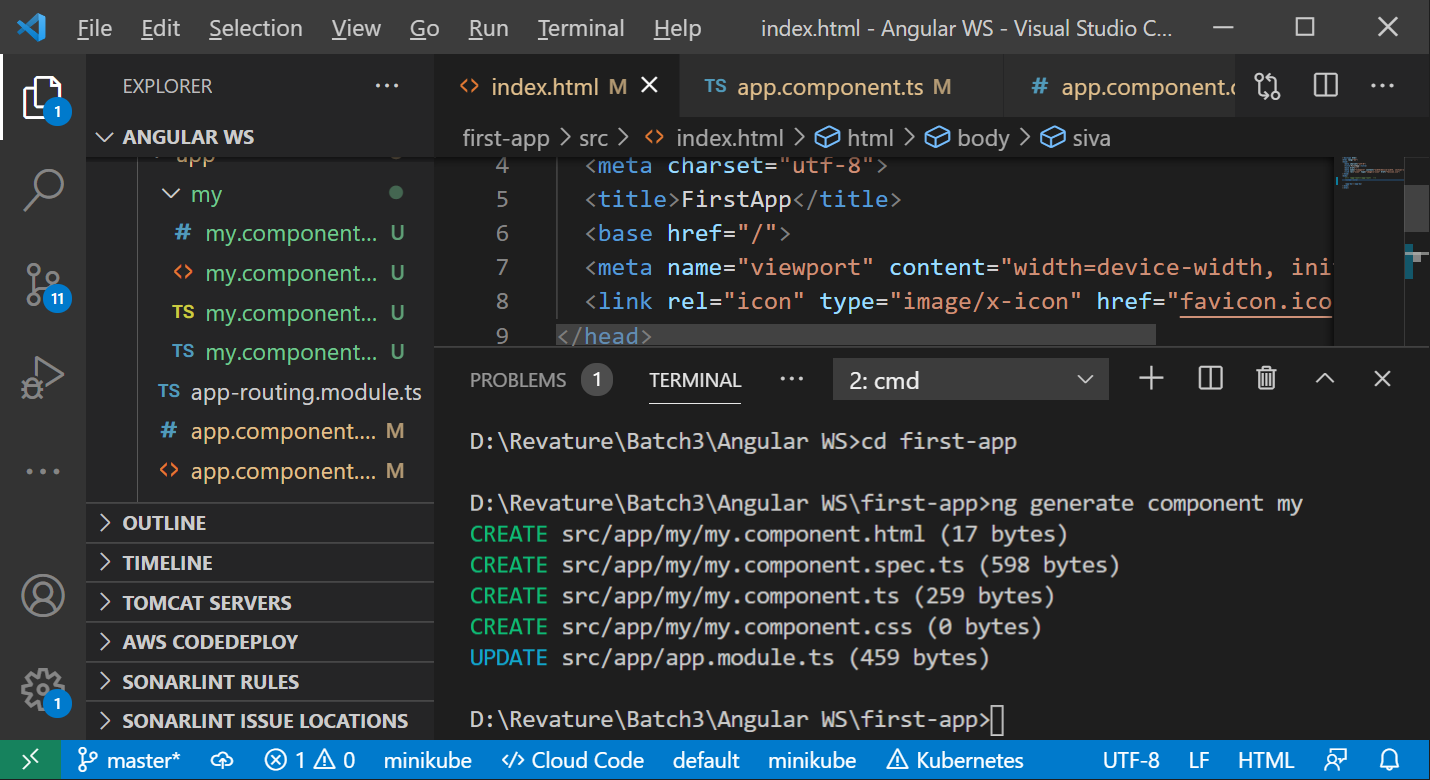
1. Interpolation Binding {{propertyName from ts file}} -- used in HTML page [From ts to HTML]
2. Property Binding[] [property name from ts file] – used in HTML page or tag attributes [ from ts to HTML]
3. Event Binding() [Event data from html file] – used in HTML tag attributes [ From HTML to ts file]
4. Two way binding [Banana in box syntax – [()]) – from html to ts & from ts to html

Components – Reusable piece of code (combination of html,css&ts)

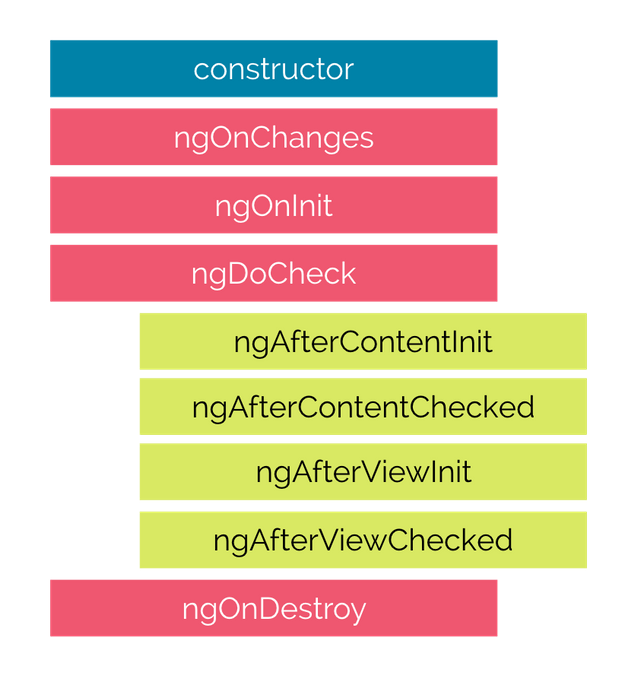
.ts – typescript file

.spec.ts – unit test code for the .ts file

ng generate component my ( ng g c my)



Life Cycle Methods



Component with Inline style and inline template

import { Component } from ‘@angular/core’;

@Component ({

selector: 'app-root',

template: `

<h3> Hello World</h3>

<p> {{title}} app is running... </p>

` ,

styles: ['h3:{ background-color : red;}', 'p{font-weight:bold}']

})

export class AppComponent {

title = 'myfirstapp';

}

Directives – starts with ng [\*ngIf, \*ngFor, \*ngSwitch … ]

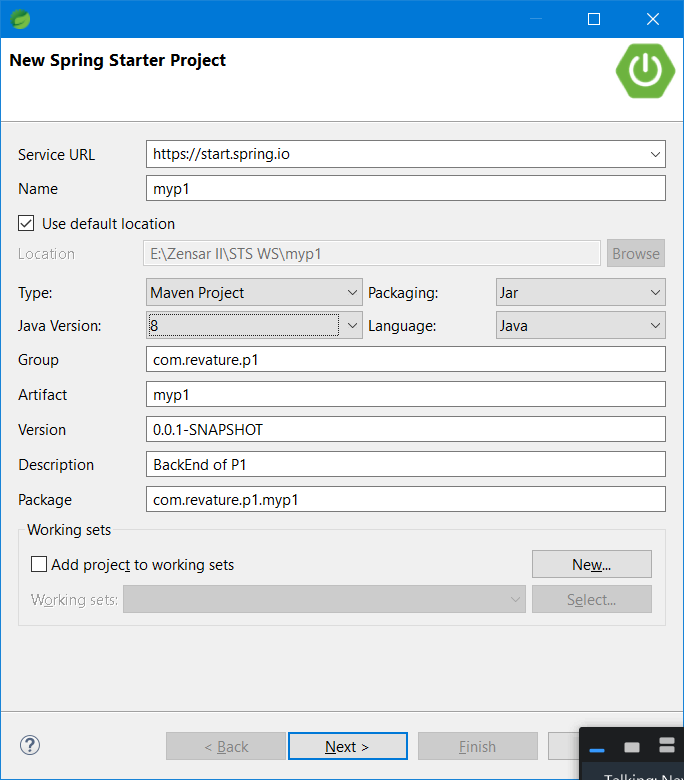
STS – Spring Tool Suite ( spring.io/tools)

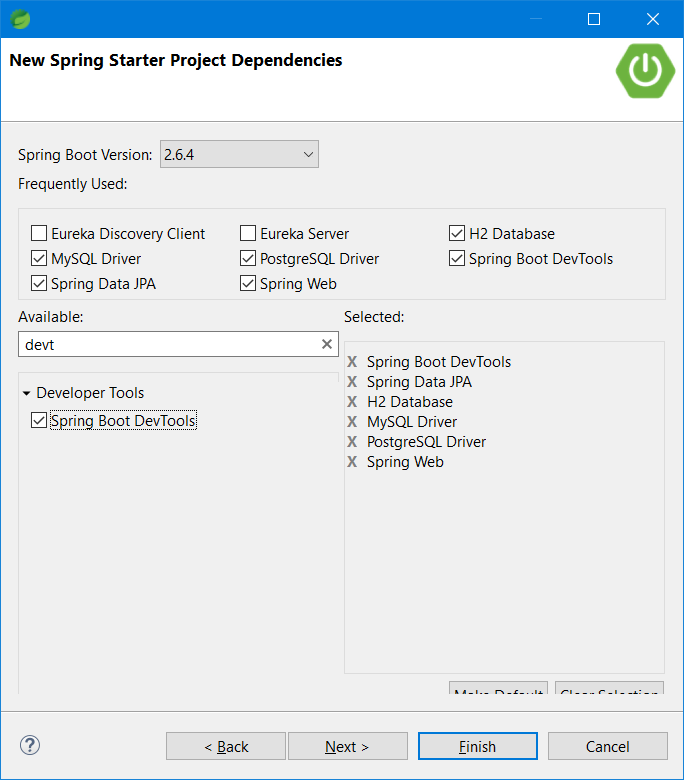
STS – Back End Coding (Web Services / Micro Services)

VS Code – Front End Development (HTML/CSS/JS or Angular )

In STS

File 🡪 New 🡪 Spring Starter Project





URI – Uniform Resource Identifier --- <http://localhost:8090/welcome>

URI === REST End point

Web Service - - Generates the result in JSON format [{"id":100,"username":"abc","password":"abc1234","email":"abc@gmail.com","mobile":8978675667}]

@RequestMapping – is a Generic Annotation (It will work with all the HTTP methods)

HTTP Method

1. Get -- reading data (@GetMapping)
2. Post - inserting /creating data (@PostMapping)
3. Put - updating the data (@PutMapping)
4. Delete – removing the data (@DeleteMapping)