Youtube videos :: angular 7  from codeevolution, design course

Functional progming

observables

/bootsnipp.com/  for bootstraps egs

Add in envt variable  : C:\Users\743639\AppData\Roaming\npm

Install :  npm install –g @angular/cli

Ng commands

Ng new <your comp name>

Ng serve  get the serverup on locahost4200. In this case the deployable is created in the memory during runtime.thuis making the deployable uneditable

Ng build  creates a deployable file in dist folder. Here the deplable is editable

Ng build –prod  creates a smaller deployable file

ng generate component family/grand-parent  generate component with a folder structure

ng generate pipe <my-pipe-name>

Directives

\*ngFor : for loop    <li \*ngFor= "let task of myTasks"> {{task}} </li>

ngModel :

Files in angular

Package.json  jasmine – for unit testcase

                                Karma –to run the unit test

                                Istanbul –test coverage  (in java its jmeter)

Angular.json (has all config details)

                                prefix :”app”

                                Outputpath : dist/angular01  (the deployable files get created here)

                                Index: src/index.hhtl (the default file that renders when the code is deplyed)

                                Main :src/main.ts (entry point for anglar prgm)

                                Styles :{…} (default style file is kept here)

Node\_modules has all the dependent files. It will get created when we do npm install cmd. It reads the package.json and download the dependencies

Src **app :** default component. It has 4 default files . .ts file, css file, html file,spec.ts

**Assets** : images,media etc are to be kept here

          Index,html :  it has a component <app-root></app-root>

          Main.ts entry point. It will do all the initializations.the following line is imp

import { AppModule } from './app/app.module'; //import the root module

platformBrowserDynamic().bootstrapModule(AppModule) // bootstraps the rootmodule

 this is the root module. It defines the root component which is referred by index.htm

(app folder)

App.module.ts 

@NgModule({

  declarations: [

    AppComponent

  ],            list of the components are given here

  providers: [], list of services

imports: [

]  any custom things

App.componet,ts  @Component   the root element, the css and the html are defined

App.componemt.spec.ts for writing testcases

Babel does the traspilation of TS to JS for browsers. Babel is present in the angular framework and managed by webtec

Single page application works with routers.

Helps to load multiple components on screen based on what user clicks..

1.       Add --import {RouterModule} from '@angular/router';  in app.componet.ts also import it

2.       Include which view the routerlinks will load as below: (in app.componet.ts)

  RouterModule.forRoot ([

                {  path :'family',

                  component: GrandParentComponent

                },

                 {  path :'task',

                  component: TaskManagerComponent

                }

Here we can have child relationships also..

path :'family',

                  component: GrandParentComponent,

                  children:[…define children componets in the same way…]

3.       Add the routerlink in the html as below

<a routerLink="/family"> Family</a>

<a routerLink="/task"> Task</a>

<router-outlet></router-outlet>

We need to add router module in the app.module.ts   :::import { AppRoutingModule } from './app-routing.module';

Pipes

Its used for filtering or modifying the data that is displayed

Filters : filer () is similar to java8. It parses one json obj at a time

Json server

Install : npm install -g json-server

Create a json file with the json objs

Run the file on server with cmd: json-server users.json

http services

Import: import {HttpClientModule} from '@angular/common/http';

this.user.getRemoteUserData().subscribe( (respData)=>{console.log(respData)}) call in the component calls and subscribes to the service. Subscribe allows us to use the returned data from the service. This Function can access the returned data.

**Note:**we use arrow function in case of subscription. Nomal function cannot be used here. Arrow function allow use to process asynchronous communication

Single page routing

Add this in root module : path :'remoteData/:userId',  where userId will be dynamic based on the user’s selection

The changes are made in component as below::

  We do dependency injection of the router in the main component   constructor(private user :RestApiUsersServiceService, private router: Router)

this.router.navigate(["/remoteData",id])  // it will navigate to remoteData/id..whatever id is passed

forms

2 types: template and reactive

formGroup – the whole form

formControl- each component of the form

its used in reactive forms. Reactive forms requires pragmatic initialization of the formGroup and formcontrol

reactive forms help to add validations for forms

angular animation

imports required :   import { AnimationComponent } from './animation/animation/animation.component';  in root module

import {trigger,state,transition,animate,style} from '@angular/animations'  in the .ts file where animation is required

Note::see the comments given in the code

Bootstrap

Getbootstrap.com  default css and styles can be copies and used in html

In bootstrap the whole page is divided into 12 cmpts

Npm install bootstrap

Npm install jquery popper.js

Modify ur angular.jspn file to include the css frpm bootstrap

"styles": [

              "src/styles.css",

              "./node\_modules/bootstrap/dist/css/bootstrap.css"

            ],