<https://www.youtube.com/watch?v=CUCJywIZrIo&list=PLz5rnvLVJX5VQZdTz4MTE52aKto-MoPwB&index=3>

API ref: <https://angular.io/docs/ts/latest/api/>

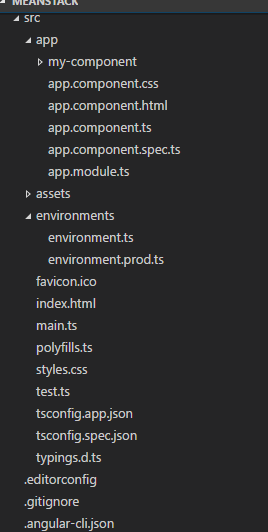
Folder in laptop: D:\Angular4\meanstack

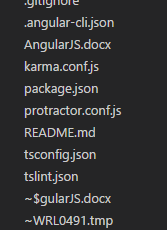
-Ensure Install npm and node are installed

-We can use the angular cli for easy installation – <http://cli-angular.io>

Npm install –g @angular/cli

Ng new mynew app





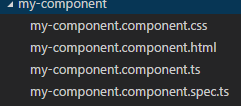
Cd mynewapp

Ng serve –host 127.0.0.1 –port 4200

Create components:

Ng g component my-component

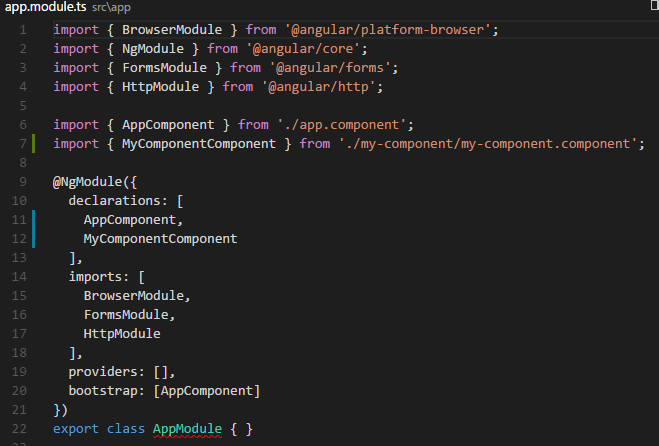
\*.spec.\* file created is a testing file



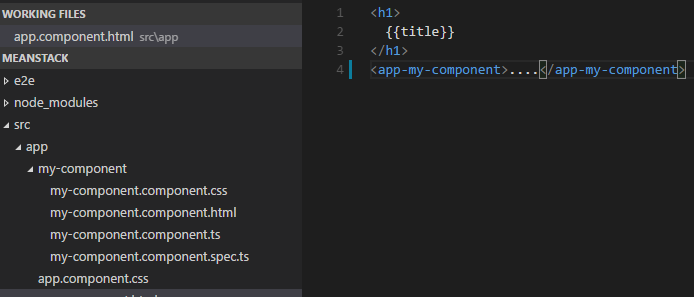
Component has a Selector, TemplateUrl



New component entry also goes in the module file🡪 app.module.ts



Add “app-my-component” tag to app.component.html to see the newly added component in the main componenent



Now browse to <http://localhost:4200/> to see the newly added component.

**Order of execution:**

Main.ts 🡪 app.module.ts 🡪 app.component.ts 🡪 my-component.componenet.ts

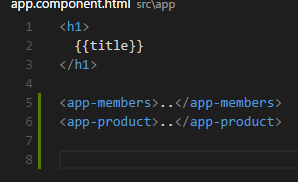
**Routing**

-Create 2 new components

Ng g component product

Ng g component members

-Add the new tags to the main component html



Browse to see the components

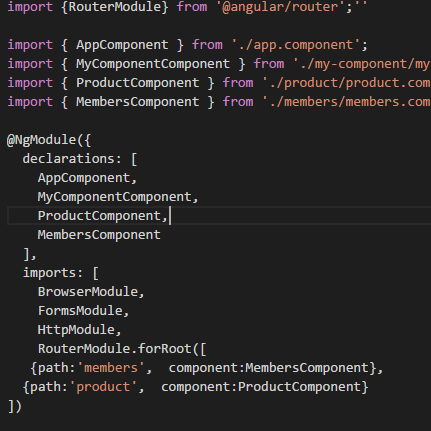
-In app.module.ts add import {RouterModule} from ‘@angular/router’

In imports section add RouterModule.forRoot([

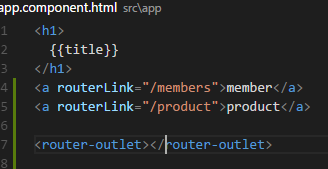
{Path: ‘member’, component:’MembersComponent’},

{Path: ‘product’, component:’ProductComponent’}

])



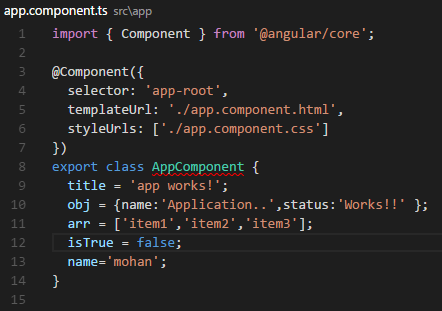
* Add links in html

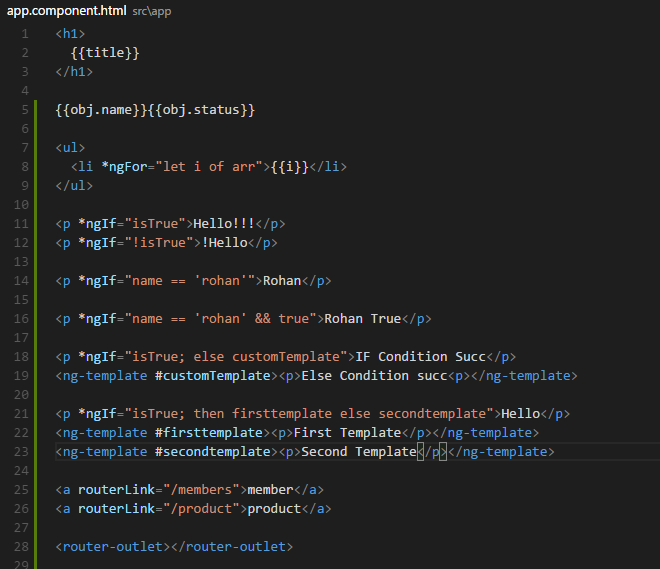


Router-outlet defines the area where the new component will be showed.

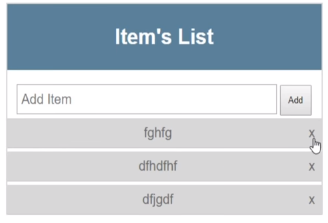
**Data Binding**

{{}} is interpolation

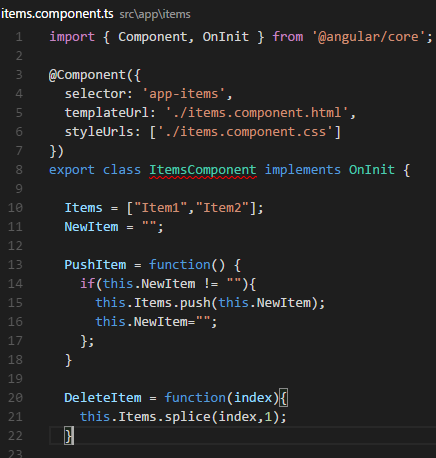




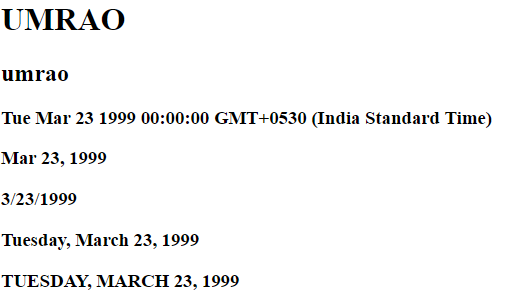
**Single Page Application**

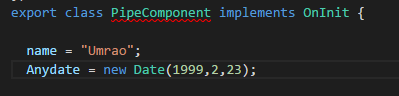


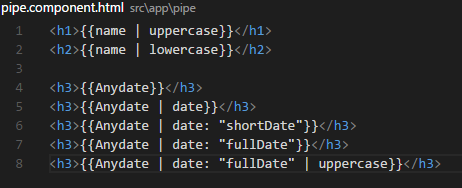




**PIPES**







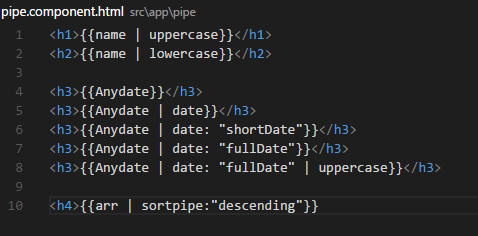
CUSTOM PIPES:

Ng g pipe sortpipe

Note: args is asc or desc









**FROMS**

Make sure all controls have name specified

TEMPLATE DRIVEN FORM

Most of work done in html template

Specify ngmodel in each control in form.

Specify (ngsubmit)=”” in the form tag.

#myForm=”ngForm” (ngSubmit)=”onSubmit(myForm.value)”



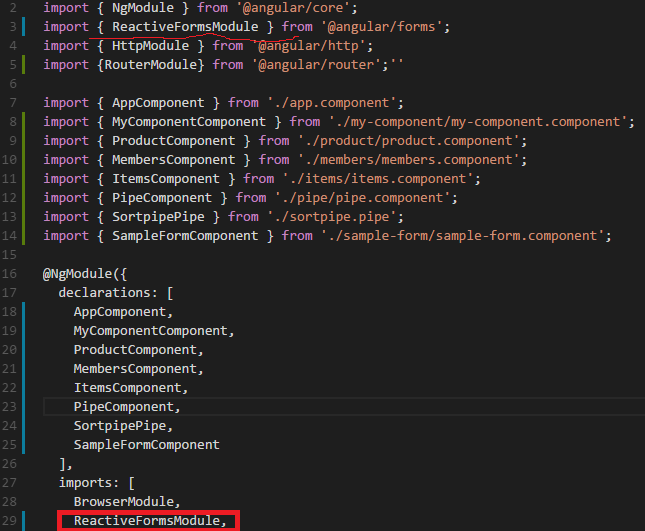


Model driven forms

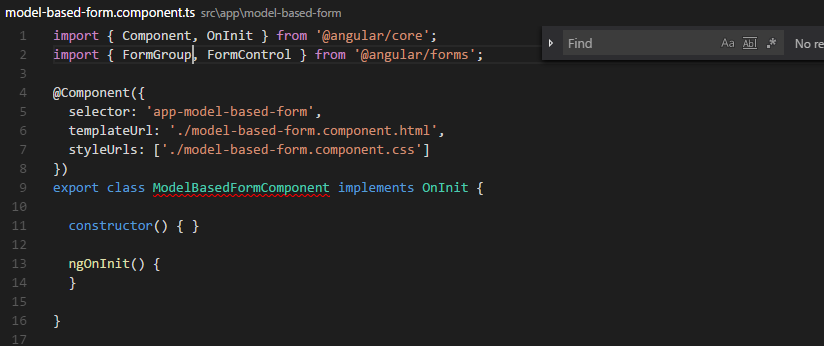
More testable and more validations can be done.

Two way data binding possible.

ReactiveFormsModule module needs to be imported.



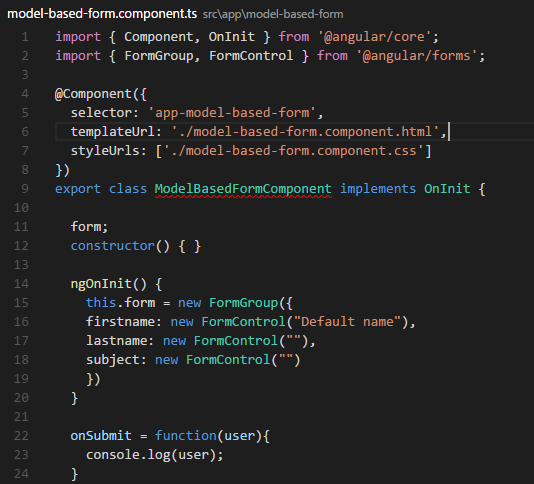
Add import FormGroup and FOrmControl



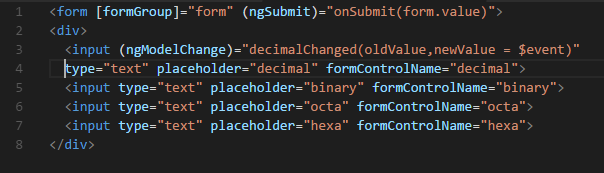
Here formControllerName is set for each control

Set the [FormGroup] in the form tag

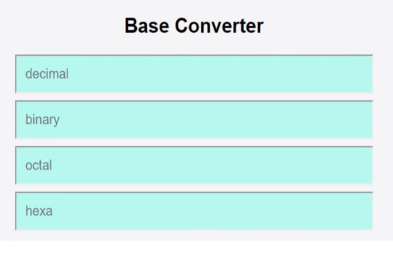




**Single Page App Full Fledged**







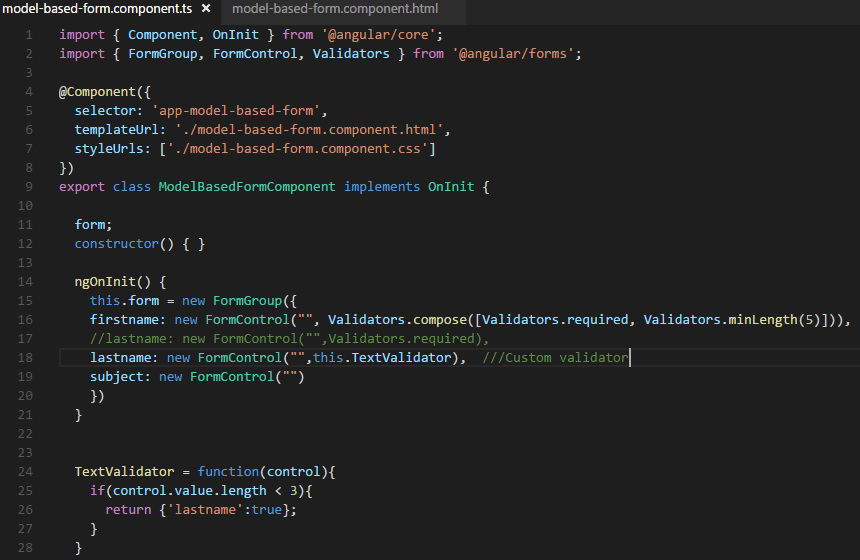
**Form Validation**

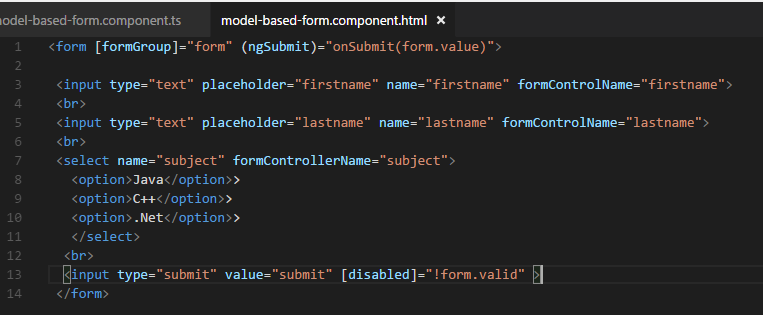
Add a entry for import {FormGroup, FormControl, Validators} from “@angular/forms”

2 ways Default validation, Custom validation

Ng-invalid class is internally applied when validations fail

Ng-valid class applied to valid controls

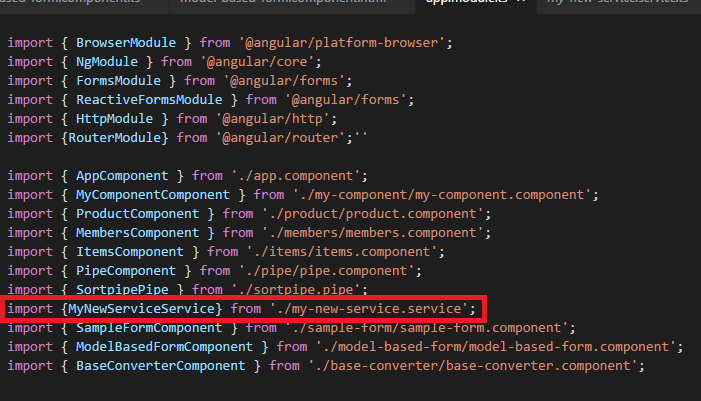


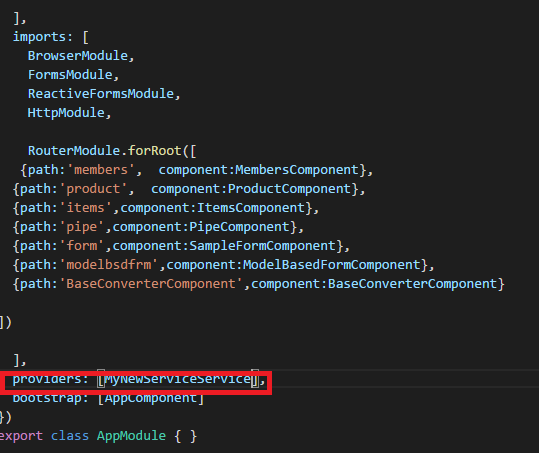


**SERVICE**

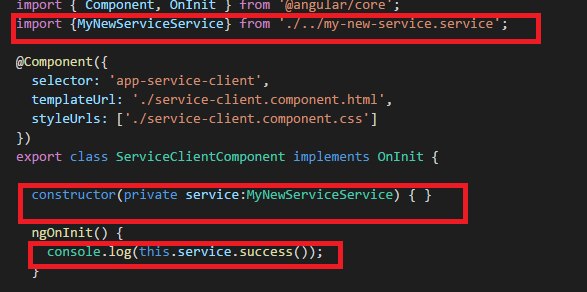
Ng g service my-new-service

Add the service to the app.module.ts as imports & in provider section

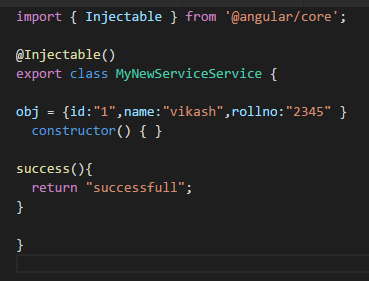




Include import of service in component .



Define functions in the service: success is user defined function



**HTTP Service**

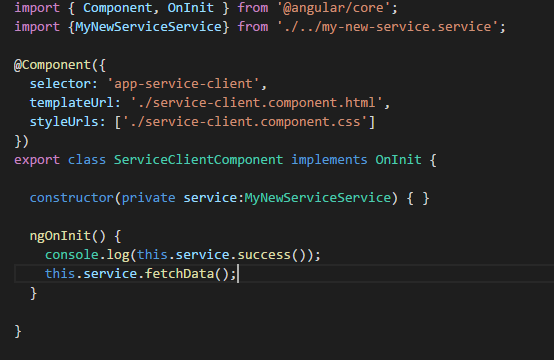
Fetch data from external source:

Import the http system module in your service.

Create private valriable http in constructor.

Student.json has the data 🡪 mimicking a service data





**Animation**

Add and install animation : npm install angular/animation@latest –save

This will also make entry to package.json