User Management App..!

1. Update npm to Latest version

npm i npm@latest -g

1. Check if old version of Angular is already installed on your computer, if exist then remove it

npm uninstall -g @angular/cli

npm cache clean

npm install -g @angular/cli

Note: To install a specific version, you can use npm install -g @angular/cli@1.4.9

1. Create new project

ng new user-management-app

or

skip to install node modules while creating project

ng new user-management-app –skip-install

npm install [run this command separately to load all dependencies]

1. Run your application to get started

cd angular6-example

ng serve

1. Create Components

ng g component home

ng g component login --spec=false

ng g component add-user

ng g component edit-user

ng g component list-user

Angular CLI Useful Commands

ng g component my-new-component

ng g directive my-new-directive

ng g pipe my-new-pipe

ng g service my-new-service

ng g class my-new-class

ng g guard my-new-guard

ng g interface my-new-interface

ng g enum my-new-enum

ng g module my-module

1. Configure Routing inside app.routing.ts

import { RouterModule, Routes } from '@angular/router';

import {LoginComponent} from "./login/login.component";

import {AddUserComponent} from "./add-user/add-user.component";

import {ListUserComponent} from "./list-user/list-user.component";

import {EditUserComponent} from "./edit-user/edit-user.component";

const routes: Routes = [

{ path: 'login', component: LoginComponent },

{ path: 'add-user', component: AddUserComponent },

{ path: 'list-user', component: ListUserComponent },

{ path: 'edit-user', component: EditUserComponent },

{path : '', component : LoginComponent}

];

export const routing = RouterModule.forRoot(routes);

// OR

Inside app-routing.module.ts file

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

// Import RouterModule

import { Routes, RouterModule } from '@angular/router';

import { LoginComponent } from './login/login.component';

import { AddUserComponent } from './add-user/add-user.component';

import { ListUserComponent } from './list-user/list-user.component';

import { EditUserComponent } from './edit-user/edit-user.component';

import { HomeComponent } from './home/home.component';

// Each route maps a URL path to a component

const routes: Routes = [

// This route specifies the route to redirect to if the path

// is empty. In our case we are redirecting to /home

// pathMatch property value can be full or prefix. For now we

// will set it to full as we want to do a full match

{ path: '',component: HomeComponent },

{ path: 'home',component:HomeComponent},

{ path: 'login', component: LoginComponent },

{ path: 'add-user', component: AddUserComponent },

{ path: 'list-user', component: ListUserComponent },

{ path: 'edit-user', component: EditUserComponent },

//{ path: '',redirectTo:'/home',pathMatch:'full' },

// {path : '', component : LoginComponent}, // or

{ path: '\*\*', component: HomeComponent }

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

1. Create a JSON server that serves the data.

npm install -g json-server

1. Run below command on terminal

json-server --watch db.json

1. Add below code inside db.json file

,

"users": [

{

"id": 1,

"firstName": "Sonu",

"lastName": "Sathyadas",

"email": "sonu@gmail.com"

},

{

"id": 2,

"firstName": "Rashmi",

"lastName": "Pawaskar",

"email": "rashmi@gmail.com"

},

{

"id": 3,

"firstName": "Smita",

"lastName": "Singh",

"email": "smita@gmail.com"

},

{

"id": 4,

"firstName": "Mahendra",

"lastName": "Shinde",

"email": "mahendra@gmail.com"

}

]

1. Run this on browser with the below given url

<http://localhost:3000/users>

1. Create new folder model and add new file user.model.ts

export class User{

id: number;

firstName: string;

lastName: string;

email: string;

}

1. Create new folder services inside app folder and then create user service by typing below command

ng g s user

and Add below code inside user.service.ts

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

import { HttpClient } from '@angular/common/http';

import { User } from '../model/user.model';

@Injectable()

export class UserService {

constructor(private http:HttpClient) { }

baseUrl:string = 'http://localhost:3000/users';

// Get All Users

getUsers(){

return this.http.get<User[]>(this.baseUrl);

}

// Get User By Id

getUserById(id: number){

return this.http.get<User>(this.baseUrl+'/'+id);

}

// Create User

createUser(user: User) {

return this.http.post(this.baseUrl, user);

}

// Modify User

updateUser(user: User) {

return this.http.put(this.baseUrl + '/' + user.id, user);

}

// Delete User

deleteUser(id: number) {

return this.http.delete(this.baseUrl + '/' + id);

}

}

1. Create new folder images inside assets folder and add one image
2. Register your services and routing here, App.module.ts file

import { BrowserModule } from '@angular/platform-browser';

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

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { LoginComponent } from './login/login.component';

import { AddUserComponent } from './add-user/add-user.component';

import { EditUserComponent } from './edit-user/edit-user.component';

import { ListUserComponent } from './list-user/list-user.component';

import { ReactiveFormsModule } from '@angular/forms';

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

import { UserService } from './services/user.service';

import { HomeComponent } from './home/home.component';

@NgModule({

declarations: [

AppComponent,

LoginComponent,

AddUserComponent,

EditUserComponent,

ListUserComponent,

HomeComponent

],

imports: [

BrowserModule,

AppRoutingModule,

ReactiveFormsModule,

HttpClientModule

],

providers: [UserService],

bootstrap: [AppComponent]

})

export class AppModule { }

1. Modify App.component.html with below code

<!--The content below is only a placeholder and can be replaced.-->

<div style="color:white; background-color: purple; padding: 10px; ">

<h2 style="text-align:center; margin-top: 20px; ">User Management Angular App ...! </h2>

</div>

<hr/>

<div class="container">

<div class="row">

<div class="col-md-8">

<router-outlet></router-outlet>

</div>

<div class="col-md-4">

<br/>

<img src="../assets/images/users.png" />

</div>

</div>

<br/><br/>

<div>

<footer><span class="text-danger"><strong>This site is copyright &copy; to LTI.

All rights are reserved here ..!</strong></span></footer>

</div>

</div>

1. Add below code inside [home.component.html](http://home.component.html) file

<br/><br/>

<div class="col-md-10">

<h3>Angular </h3>

<br/>

<h5>In this demo, we will learn about </h5>

<ul class="text-success" style="font-weight: bold">

<li>

Angular Architecture

<ul>

<li>Modules</li>

<li>Components

<ol>

<li>Root Component or App Component</li>

</ol>

</li>

<li>Services and Dependency Injection</li>

<li>Models</li>

<li>Directives</li>

<li>Pipes, etc</li>

</ul>

</li>

</ul>

</div>

<a class="text-right" [routerLink]="['/login']"><h3>Login..</h3></a>

1. Modify login.component.html file

<br/><br/>

<div class="col-md-6 col-md-offset-2">

<h3>Login</h3>

<br/>

<form [formGroup]="loginForm" (ngSubmit)="onSubmit()">

<div class="form-group">

<label for="email">Email address:</label>

<input type="email" class="form-control" formControlName="email" id="email">

<div \*ngIf="submitted && loginForm.controls.email.errors" class="error">

<div class="text-danger" \*ngIf="loginForm.controls.email.errors.required">Email is required</div>

</div>

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" formControlName="password" id="pwd">

<div \*ngIf="submitted && loginForm.controls.password.errors" class="error">

<div class="text-danger" \*ngIf="loginForm.controls.password.errors.required">Password is required</div>

</div>

</div>

<button class="btn btn-success">Login</button> &nbsp; &nbsp;

<button class="btn btn-danger" type="reset">Cancel</button>

<br/><br/>

<div \*ngIf="invalidLogin" class="error">

<div class="text-danger">Invalid credentials.</div>

</div>

</form>

</div>

1. Modify **login.component.ts file**

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

import {FormBuilder, FormGroup, Validators} from "@angular/forms";

import {Router} from "@angular/router";

@Component({

selector: 'app-login',

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

styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit {

loginForm: FormGroup;

submitted: boolean = false;

invalidLogin: boolean = false;

constructor(private formBuilder: FormBuilder, private router: Router, ) { }

onSubmit(){

this.submitted = true;

if(this.loginForm.invalid){

return;

}

if(this.loginForm.controls.email.value =="narend@gmail.com" &&

this.loginForm.controls.password.value=="123456"){

localStorage.setItem("username",this.loginForm.controls.email.value);

this.router.navigate(['list-user']);

}

else{

this.invalidLogin = true;

}

}

ngOnInit() {

this.loginForm = this.formBuilder.group({

email: ['', Validators.required],

password: ['', Validators.required]

});

}

}

1. Modify list-user.component.html

<br/><br/>

<div class="col-lg-12">

<h2> User Details</h2>

<br/>

<button class="btn btn-success" (click)="addUser()"> Add User</button> &nbsp;&nbsp;

<button class="btn btn-danger" (click)="logOutUser()"> LogOff</button>

<br/><br/>

<table class="table table-striped table-responsive table-hover">

<thead>

<tr>

<th class="text-hide">Id</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let user of users">

<td class="text-hide">{{user.id}}</td>

<td>{{user.firstName | uppercase }}</td>

<td>{{user.lastName | lowercase }}</td>

<td>{{user.email | titlecase }}</td>

<td><button class="btn btn-danger" (click)="deleteUser(user)"> Delete</button>

<button class="btn btn-warning" (click)="editUser(user)" style="margin-left: 20px;"> Edit</button></td>

</tr>

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

<th colspan="5">

<p class="text-right text-danger">Sorry No Users Available</p>

</th>

</tr>

</tbody>

</table>

</div>

1. Modify **list-user.component.ts**

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

import { User } from '../model/user.model';

import { UserService } from '../services/user.service';

import { Router } from '@angular/router';

@Component({

selector: 'app-list-user',

templateUrl: './list-user.component.html',

styleUrls: ['./list-user.component.css']

})

export class ListUserComponent implements OnInit {

users: User[];

constructor(private router: Router, private userService: UserService) { }

// Initialize with default list of users

ngOnInit() {

if(localStorage.getItem("username")!=null){

this.userService.getUsers()

.subscribe(data=> {

this.users = data;

});

}

else

this.router.navigate(['/login']);

}

// logOff user

logOutUser():void{

if(localStorage.getItem("username")!=null){

localStorage.removeItem("username");

this.router.navigate(['/login']);

}

}

// Delete User

deleteUser(user: User): void {

let result = confirm('Do you want to delete the user?')

if(result)

{

this.userService.deleteUser(user.id)

.subscribe( data => {

this.users = this.users.filter(u => u !== user);

});

}

};

// Modify USer

editUser(user: User): void {

localStorage.removeItem("editUserId");

localStorage.setItem("editUserId", user.id.toString());

this.router.navigate(['edit-user']);

};

// Add New User

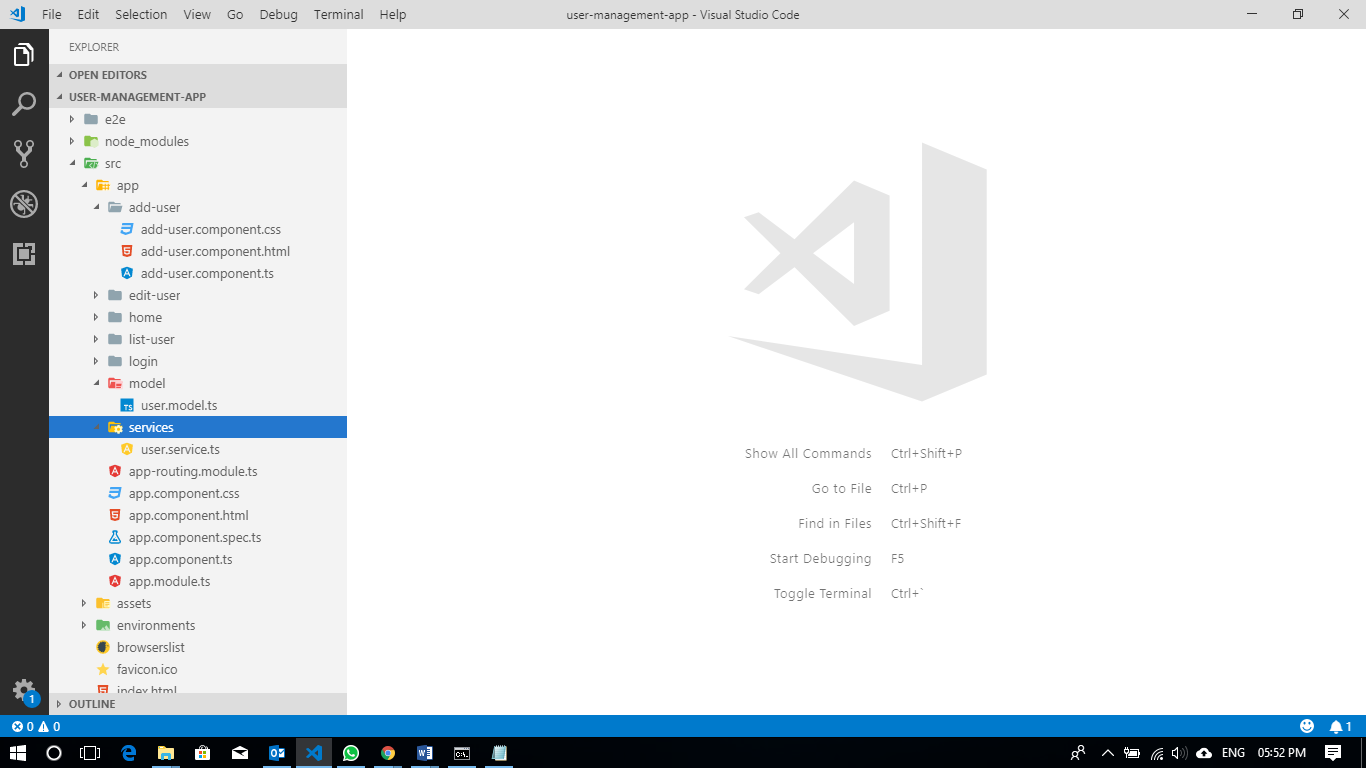
addUser(): void {

this.router.navigate(['add-user']);

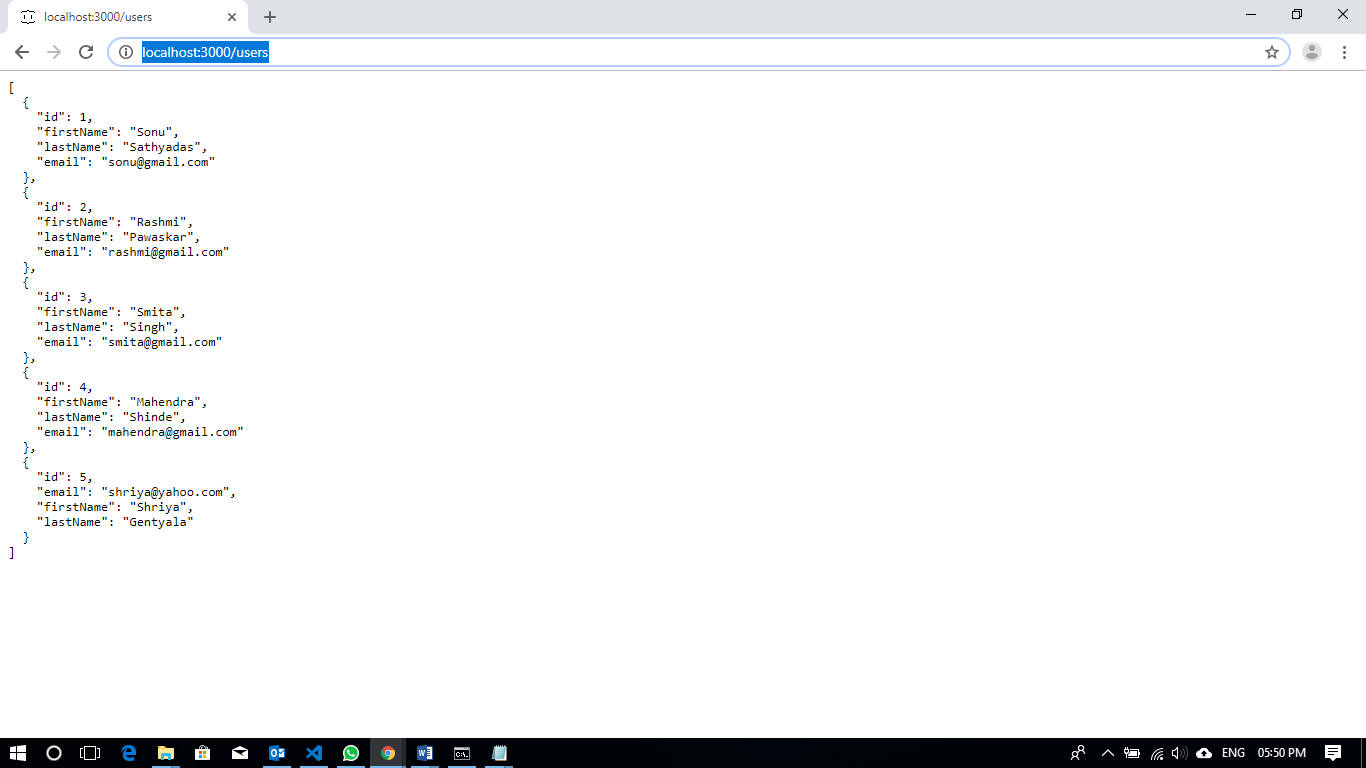
};

}

1. Testing Angular Application



json-server –watch db.json



ng serve

