### SampForm –Angular Client Project Structure

### ng new SAMPFORM

### cd SAMPFORM

### Ng g c home

### Ng g c employee-form

### Ng g c edit-user

### ng serve

### 

### Index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>SampForm</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

</head>

<body>

<app-root></app-root>

</body>

</html>

### app.component.html

<div class="container">

<router-outlet></router-outlet>

</div>

### app.module.ts

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

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

import {FormsModule} from '@angular/forms'

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

import { EmployeeFormComponent } from './employee-form/employee-form.component';

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

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

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

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

import { HttpModule } from '@angular/http';

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

import { ModalModule } from 'ngx-bootstrap';

*//enableProdMode();*

const appRoutes:**Routes**=[

{

path:'',

component:HomeComponent

},

{

path:'emp',

component:EmployeeFormComponent

},

{

path:'edituser/:id',

component:EditUserComponent

}];

@**NgModule**({

declarations: [

AppComponent,

EmployeeFormComponent,

HomeComponent,

EditUserComponent

],

imports: [

BrowserModule,

HttpModule,

FormsModule,

AppRoutingModule,

RouterModule.**forRoot**(appRoutes),

ModalModule.**forRoot**()

],

providers: [],

bootstrap: [AppComponent]

})

export class **AppModule** { }

### emp.ts

export class emp {

id:number;

name:string;

age:number;

dob:string;

degree:string;

department:string;

grade:string;

constructor(

id:number,

name:string,

age:number,

dob:string,

degree:string,

department:string,

grade:string){}

}

### Ng g c home.componenet

### home.component.html

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

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<h4>New Employee Register

<!-- <a routerLink="emp" routerLinkActive="active" >here</a> -->

<button align="right" class="btn btn-primary" (click)="empform()" value="Add User">Add User</button>

</h4>

<h2> Emp Details</h2>

<table class="table table-striped">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>age</th>

<th>dob</th>

<th>degree</th>

<th>department</th>

<th>grade</th>

<th>Edit</th>

<th>delete</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let e of empData">

<td>{{e.id}}</td>

<td>{{e.name}}</td>

<td>{{e.age}}</td>

<td>{{e.dob}}</td>

<td>{{e.degree}}</td>

<td>{{e.department}}</td>

<td>{{e.grade}}</td>

<td>

<button class="btn btn-primary" (click)="updateUser(e)" >Edit</button>

</td>

<td>

<button class="btn btn-primary" (click)="openModal(template)"> Delete</button>

</td>

<!-- <button class="btn-large btn-danger" (click)="updateUser(e)" style="margin-left: 20px;"> Edit</button></td> -->

<ng-template #template>

<div class="modal-body text-center">

<p>Do you want to confirm?</p>

<button type="button" class="btn btn-default" (click)=" deleteUser(e)" >Yes</button>

<button type="button" class="btn btn-primary" (click)="decline()" >No</button>

</div>

</ng-template>

</tr>

</tbody>

</table>

</div>

### home.component.ts

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

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

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

import {emp} from '../home/emp';

import { Http , Response } from '@angular/http';

import {EmployeeserviceService} from '../employeeservice.service';

import { Employee } from '../shared/emp.model';

import { BsModalService } from 'ngx-bootstrap/modal';

import { BsModalRef } from 'ngx-bootstrap/modal/bs-modal-ref.service';

@Component({

selector: 'app-home',

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

styleUrls: ['./home.component.css'],

providers: [EmployeeserviceService]

})

export class HomeComponent implements OnInit {

empData: emp[];

e:emp;

EmpeditForm:NgForm;

bsmodelref: BsModalRef;

message: string;

constructor(

private router:Router,

private \_eventData: EmployeeserviceService,

private bsmodel: BsModalService) { }

ngOnInit() {

this.\_eventData .getuserdata().subscribe(

data=> { this.empData=data });

}

openModal(template: TemplateRef<any>) {

this.bsmodelref = this.bsmodel.show(template, {class: 'modal-sm'});

}

empform(){

this.router.navigate(['emp']);

}

deleteUser(e:emp) {

//console.log('sss'+e.id);

this.\_eventData.deleteUser(e.id)

.subscribe(data => {this.empData.filter(e1=>e1!==e) ;

console.log(data);

this.ngOnInit();

this.bsmodelref.hide();

},

error => console.log(error));

}

decline(): void {

this.message = 'Declined!';

this.bsmodelref.hide();

}

updateUser(e){

console.log(e);

localStorage.removeItem("editUserId");

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

this.router.navigate(['edituser/'+e.id]);

}

}

### employeeservice.service.ts

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

import {Response,URLSearchParams,RequestOptions } from '@angular/http';

import { Observable } from 'rxjs/Observable';

import 'rxjs/add/operator/map';

import 'rxjs/add/operator/do';

import 'rxjs/add/operator/catch';

import { emp } from './home/emp';

import { Location } from '@angular/common';

import { Headers, Http } from '@angular/http';

import 'rxjs/add/operator/toPromise';

@Injectable({

providedIn: 'root'

})

export class EmployeeserviceService {

private headers = new Headers({

});

baseUrl: string = 'http://localhost:8080/SpringRestHibernate';

constructor(private \_http: Http) { }

postuserdata(emp:Object) {

console.log(emp);

return this.\_http.post(this.baseUrl+'/create',emp);

}

getuserdata(): Observable<emp[]> {

return this.\_http.get(this.baseUrl+'/students')

.map(response => response.json() as emp[])

.catch(this.handleError);

}

deleteUser(id: number): Observable<any> {

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

}

getUser(id): Observable<emp> {

return this.\_http.get(this.baseUrl+'/student/'+id)

.map(response => response.json() as emp)

.catch(this.handleError);

}

updateUser(emp:Object) {

console.log('\*\*\*\*\*\*\*\*\*\*\*'+emp);

return this.\_http.put(this.baseUrl + '/update',emp);

}

private handleError(error: Response) {

console.error(error);

return Observable.throw(error.json().error());

}

}

### Ng g c employee-form

### employee-form.component.html

<div class="row ">

<div class="col s8 offset-s2">

<div class="card ">

<div class="sm-jumbotron center-align">

<h2 class="text-center">Add User</h2>

<form name="EmpForm" action="" method="post" class="col.s12.white" #EmpForm="ngForm">

<div class="row" >

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

<label>Id</label>

<input type="number" class="form-control" name="id" id="id" [(ngModel)]="employee.id" #id="ngModel" required >

<div \*ngIf="id.invalid && (id.dirty || id.touched)"

class="alert alert-danger">

<div \*ngIf="id.errors.required">

id is required.

</div>

</div>

</div>

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

<label>Name</label>

<input type="text" class="form-control" name="name" id="name" [(ngModel)]="employee.name" required>

</div>

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

<label>Age</label>

<input type="number" class="form-control" name="age" id="age" [(ngModel)]="employee.age" required>

</div>

</div>

<div class="row" >

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

<label>DOB</label>

<input type="text" class="form-control" name="dob" id="dob" [(ngModel)]="employee.dob" required>

</div>

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

<label>degree</label>

<input type="text" class="form-control" name="degree" id="degree" [(ngModel)]="employee.degree" required>

</div>

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

<label>department</label>

<input type="text" class="form-control" id="department" name="department" [(ngModel)]="employee.department" required>

</div>

</div>

<div class="row" >

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

<label>grade</label>

<input type="text" class="form-control" name="grade" id="grade" [(ngModel)]="employee.grade" required>

</div>

</div>

<div class="row" >

</div>

<div class="row" >

<div class="form-group col-md-12 text-center">

<!-- login(EmpForm.value)-->

<button class="btn btn-default" (click)="openModal(template)" >Submit</button>

<ng-template #template>

<div class="modal-body text-center">

<p>Record saved !</p>

<button type="button" class="btn-large btn-submit" type="submit" (click)="login()" >Ok</button>

</div>

</ng-template>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

### employee-form.component.ts

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

import { Employee } from '../shared/emp.model';

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

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

import {EmployeeserviceService} from '../employeeservice.service';

import {emp} from '../home/emp';

import { BsModalService } from 'ngx-bootstrap/modal';

import { BsModalRef } from 'ngx-bootstrap/modal/bs-modal-ref.service';

@Component({

selector: 'app-employee-form',

templateUrl: './employee-form.component.html',

styleUrls: ['./employee-form.component.css'],

providers: [EmployeeserviceService]

})

export class EmployeeFormComponent implements OnInit {

empData: emp[];

bsmodelref: BsModalRef;

//EmpForm : FormGroup;

// Employee=new Employee();

employee: Employee=new Employee();

model:any;

constructor(private router:Router,private bsmodel: BsModalService,private \_empData: EmployeeserviceService) {

//private formBuilder: FormBuilder

} ngOnInit() {

}

login(){

this.\_empData.postuserdata(this.employee).subscribe(data =>{

console.log(data);

this.bsmodelref.hide();

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

});

}

openModal(template: TemplateRef<any>) {

this.bsmodelref = this.bsmodel.show(template, {class: 'modal-sm'});

}

}

### employee-form.component.spec.ts

import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { EmployeeFormComponent } from './employee-form.component';

describe('EmployeeFormComponent', () => {

let component: EmployeeFormComponent;

let fixture: ComponentFixture<EmployeeFormComponent>;

beforeEach(async(() => {

TestBed.configureTestingModule({

declarations: [ EmployeeFormComponent ]

})

.compileComponents();

}));

### Ng g c edit-user

### edit-user.component.html

<div class="row ">

<div class="col s8 offset-s2">

<div class="card ">

<div class="sm-jumbotron center-align">

<h2 class="text-center">Edit User</h2>

<form name="EmpeditForm" action="" method="post" class="col.s12.white" #EmpeditForm="ngForm">

<div class="row" >

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

<label>Id</label>

<input type="number" class="form-control" name="id" id="id" [(ngModel)]="empData.id" required>

</div>

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

<label>Name</label>

<input type="text" class="form-control" name="name" id="name" [(ngModel)]="empData.name" required>

</div>

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

<label>Age</label>

<input type="number" class="form-control" name="age" id="age" [(ngModel)]="empData.age" required>

</div>

</div>

<div class="row" >

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

<label>DOB</label>

<input type="text" class="form-control" name="dob" id="dob" [(ngModel)]="empData.dob" required>

</div>

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

<label>degree</label>

<input type="text" class="form-control" name="degree" id="degree" [(ngModel)]="empData.degree" required>

</div>

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

<label>department</label>

<input type="text" class="form-control" id="department" name="department" [(ngModel)]="empData.department" required>

</div>

</div>

<div class="row" >

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

<label>grade</label>

<input type="text" class="form-control" name="grade" id="grade" [(ngModel)]="empData.grade" required>

</div>

</div>

<div class="row" >

<div class="form-group col-md-12 text-center">

<!-- login(EmpForm.value)-->

<button class="btn btn-default" (click)= "openModal(template)">Update</button>

<ng-template #template>

<div class="modal-body text-center">

<p>Record Updated !</p>

<button type="button" class="btn-large btn-submit" type="submit" (click)="update()" >Ok</button>

</div>

</ng-template>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

### edit-user.component.ts

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

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

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

import {EmployeeserviceService} from '../employeeservice.service';

import {emp} from '../home/emp';

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

import { Employee } from '../shared/emp.model';

import { BsModalService } from 'ngx-bootstrap/modal';

import { BsModalRef } from 'ngx-bootstrap/modal/bs-modal-ref.service';

import { ActivatedRoute, Params ,} from '@angular/router';

@Component({

selector: 'app-edit-user',

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

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

})

export class EditUserComponent implements OnInit {

bsmodelref: BsModalRef;

empData: emp;

employee: Employee=new Employee();

editId:string;

model:any;

constructor(

private router: Router,

private route: ActivatedRoute,

private bsmodel: BsModalService,

private \_empData: EmployeeserviceService) {

this.editId=this.route.snapshot.paramMap.get("id");

console.log(this.editId);

this.empData=new emp(0,'',0,'','','','');

}

ngOnInit(): void

{

this.\_empData.getUser(this.editId)

.subscribe( data => {

this.empData=data

});

}

update(){

this

.\_empData.updateUser(this.empData).subscribe(res => console.log('Done'));

this.bsmodelref.hide();

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

}

openModal(template: TemplateRef<any>) {

this.bsmodelref = this.bsmodel.show(template, {class: 'modal-sm'});

}

}

### package.json

{

"name": "samp-form",

"version": "0.0.0",

"scripts": {

"ng": "ng",

"start": "ng serve",

"build": "ng build",

"test": "ng test",

"lint": "ng lint",

"e2e": "ng e2e"

},

"private": true,

"dependencies": {

"@angular/animations": "^6.0.2",

"@angular/common": "^6.0.2",

"@angular/compiler": "^6.0.2",

"@angular/core": "^6.0.2",

"@angular/forms": "^6.0.2",

"@angular/http": "^6.0.2",

"@angular/platform-browser": "^6.0.2",

"@angular/platform-browser-dynamic": "^6.0.2",

"@angular/router": "^6.0.2",

"core-js": "^2.5.4",

"ngx-bootstrap": "^3.0.0",

"rxjs": "^6.0.0",

"rxjs-compat": "^6.2.0",

"zone.js": "^0.8.26"

},

"devDependencies": {

"@angular/compiler-cli": "^6.0.2",

"@angular-devkit/build-angular": "~0.6.3",

"typescript": "~2.7.2",

"@angular/cli": "~6.0.3",

"@angular/language-service": "^6.0.2",

"@types/jasmine": "~2.8.6",

"@types/jasminewd2": "~2.0.3",

"@types/node": "~8.9.4",

"codelyzer": "~4.2.1",

"jasmine-core": "~2.99.1",

"jasmine-spec-reporter": "~4.2.1",

"karma": "~1.7.1",

"karma-chrome-launcher": "~2.2.0",

"karma-coverage-istanbul-reporter": "~1.4.2",

"karma-jasmine": "~1.1.1",

"karma-jasmine-html-reporter": "^0.2.2",

"protractor": "~5.3.0",

"ts-node": "~5.0.1",

"tslint": "~5.9.1"

}

}

### Reffered:

<https://www.devglan.com/angular/angular-6-example>

[**https://github.com/only2dhir/angular6-example/tree/master/src/app/model**](https://github.com/only2dhir/angular6-example/tree/master/src/app/model)

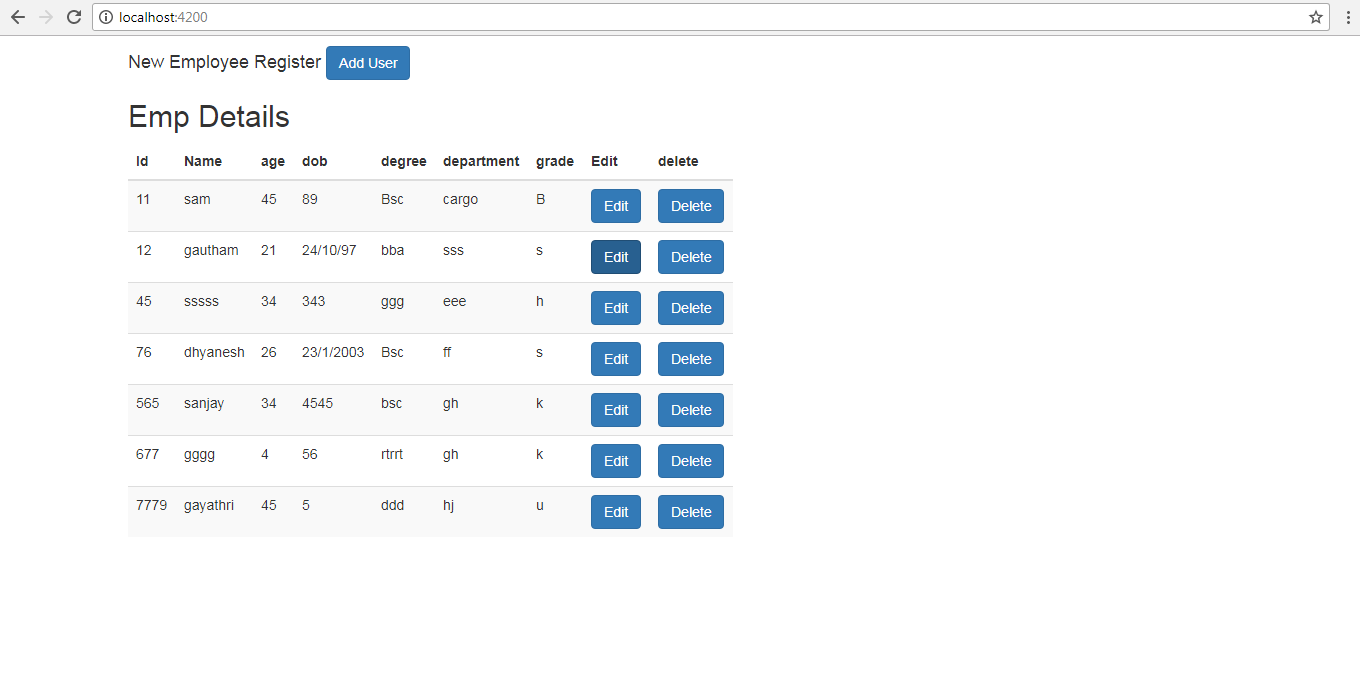
**Confirm Window –**<https://valor-software.com/ngx-bootstrap/#/modals>

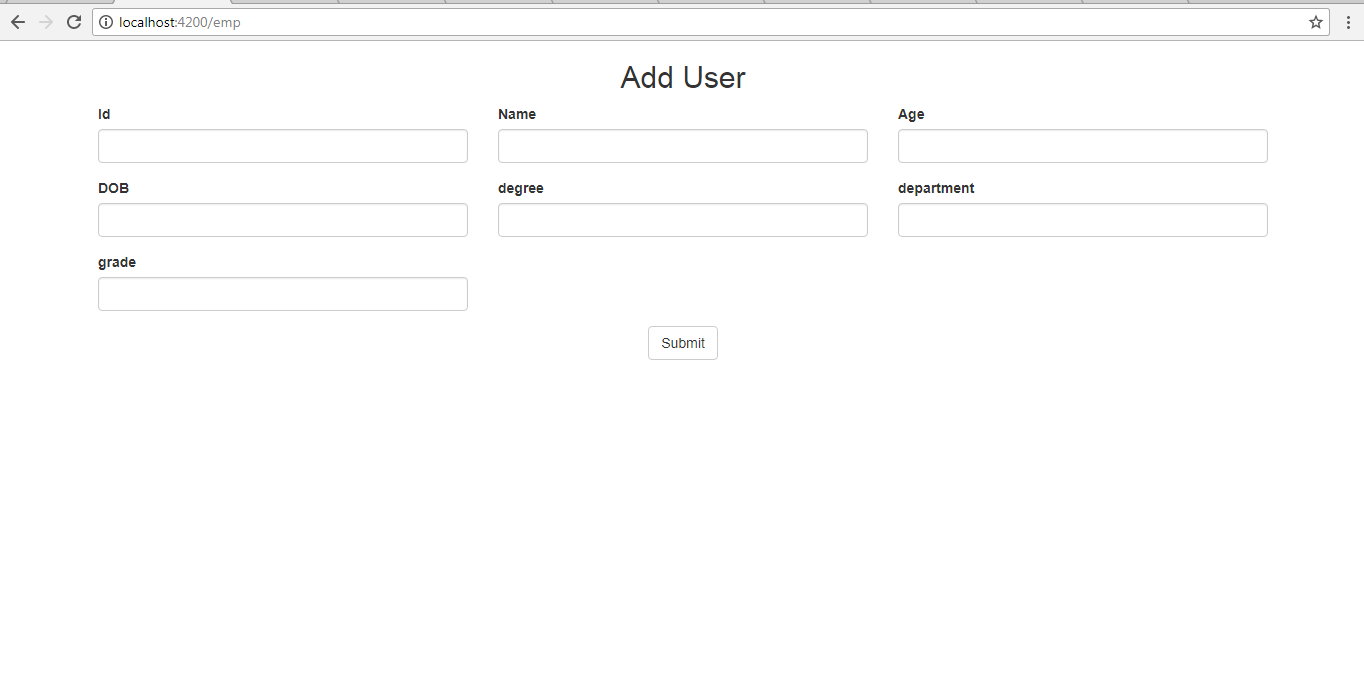
**npm install ngx-bootstrap**

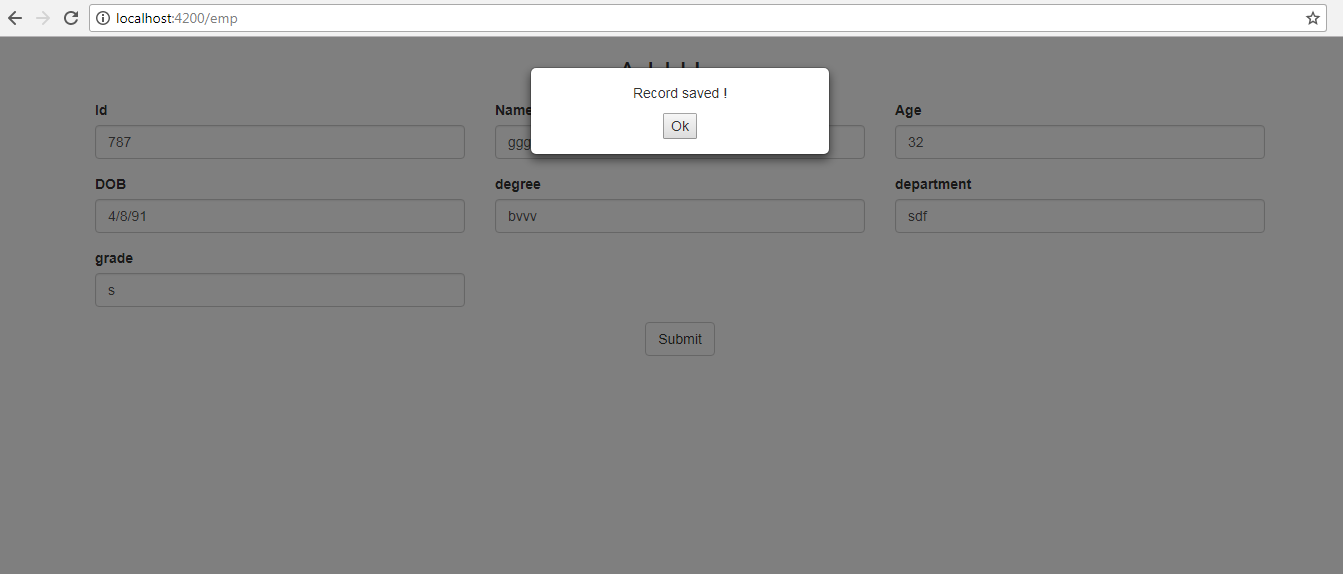
Template Form Validation :

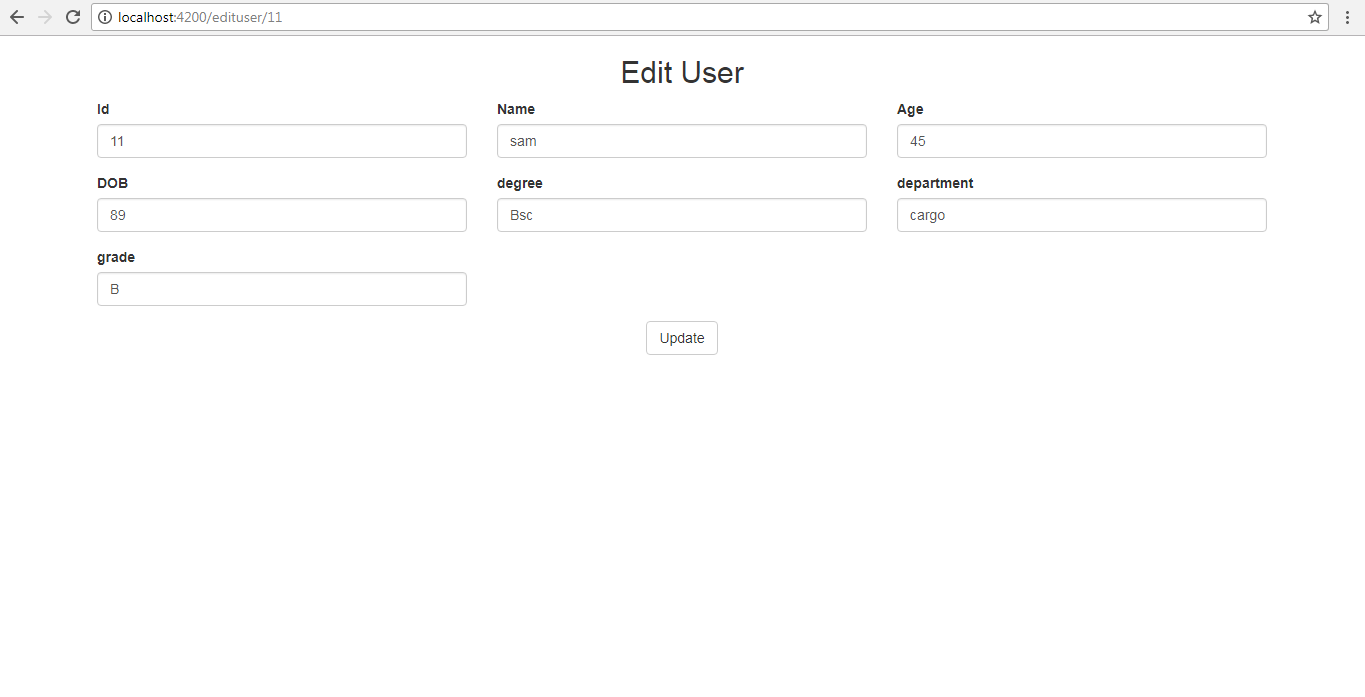
<https://angular.io/guide/form-validation>

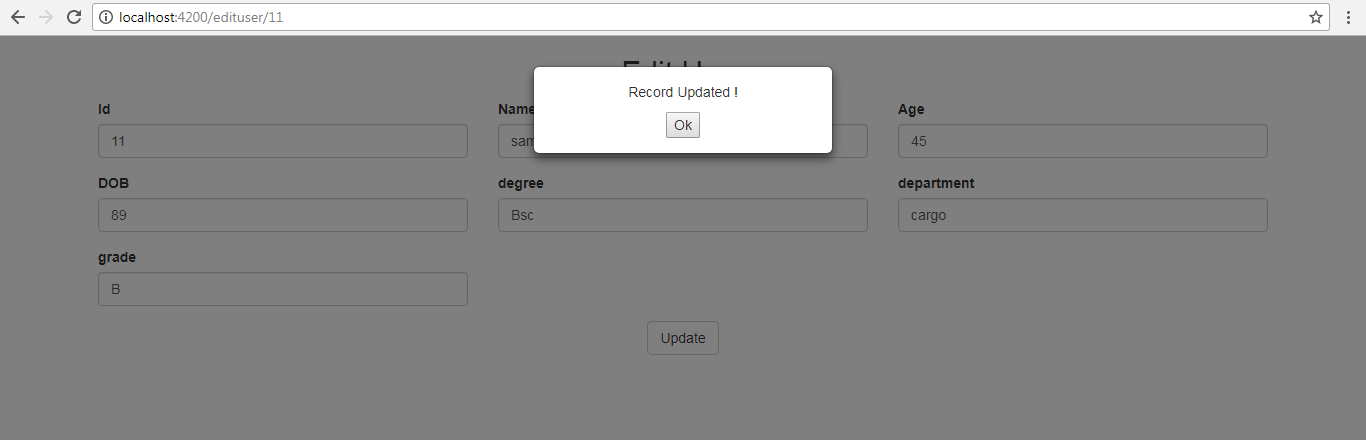
**To run** **Ng serve** After **http://Localhost:4200**

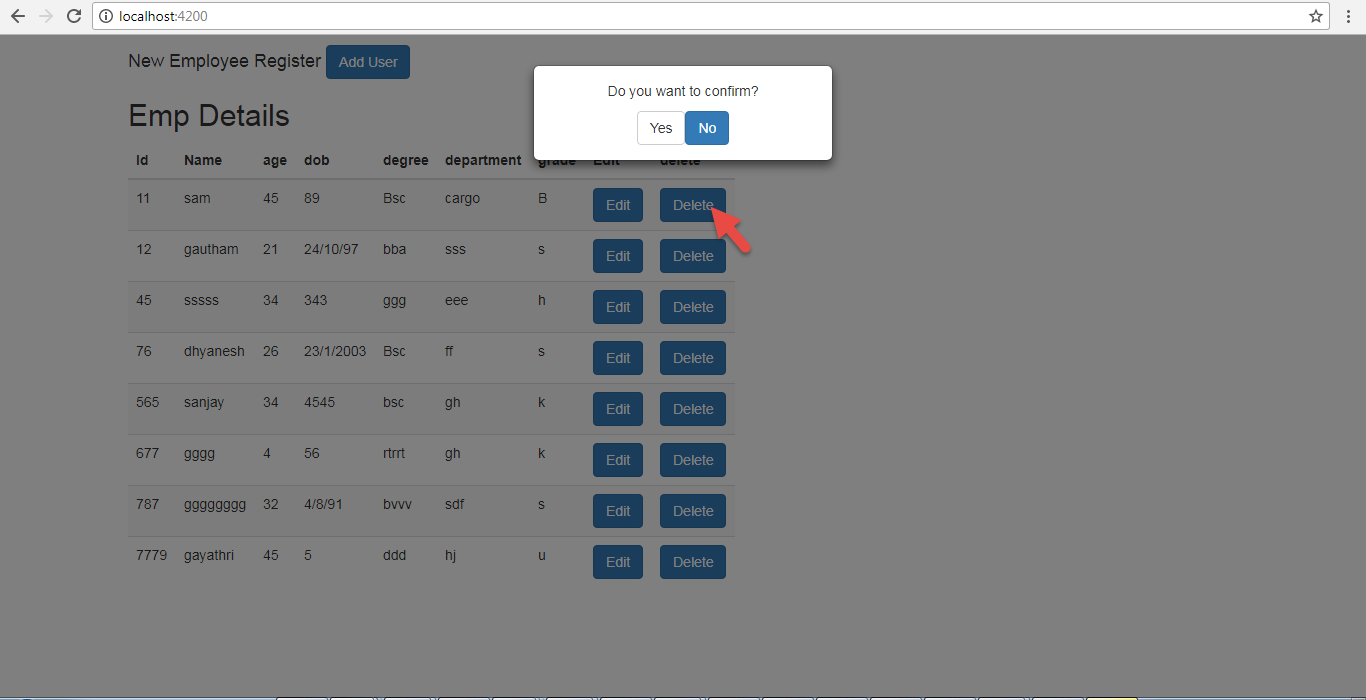












Validation syntax:

### <input id="name" name="name" class="form-control" required [minlength](https://angular.io/api/forms/MinLengthValidator)="4" appForbiddenName="bob" [([ngModel](https://angular.io/api/forms/NgModel))]="hero.name" #name="[ngModel](https://angular.io/api/forms/NgModel)" > <div \*[ngIf](https://angular.io/api/common/NgIf)="name.invalid && (name.dirty || name.touched)" class="alert alert-danger"> <div \*[ngIf](https://angular.io/api/common/NgIf)="name.errors.required"> Name is required. </div>

### <https://github.com/sujathaananad/sampformangularCRUD.git>

### https://github.com/sujathaananad/spring-REST-API-.git