Step 1: install the Node JS (<https://nodejs.org/en/download>)

Test: in Command prompt check the command like **“npm -v”**

Step 2: install the visual studio (<https://code.visualstudio.com/download>)

Step 3: install the angular (please refer the URL =><https://angular.io/cli>)

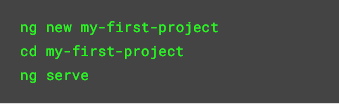
**npm install -g @angular/cli**

OR

**npm I -g @angular/cli**

Test: In command prompt check the command like **“ng version”**

Step 4 : create the angular application.



Here my-first-project is project name

**Note:** While creating the angular project select router has “yes” and Select CSS for stylesheet

Step 4.1 : ng new projectName => create the project

Step 4.2 : cd projectName🡺 open the folder

Step 4.3: ng serve 🡺 to start the application.

For start the application you can use below commands also

ng serve --o

ng serve --open

Step 5: adding bootstap in angular application

ng add @ng-bootstrap/ng-bootstrap

Step 6: adding material in angular application

ng add @angular/material

Step 7: Add the Httpclientmodule and FormsModule in app.module.ts

Step 8 : create the components

* + Header
  + Footer
  + Employee-list
  + Create-emp
  + Employee-update
  + Employee-details
  + PageNotFound

Step 9: create the class (Employee class)

export class Employee {

        id!:number

        name!:string

        address!:string

        gender!:String

        salary!:number

}

Step 10: create router for links

const routes: Routes = [

  {path:"emplist",component:EmployeeListComponent},

  {path:"update/:id",component:EmployeeUpdateComponent},

  {path:"details/:id",component:EmployeeDetailsComponent},

  {path:"create",component:CreateEmployeeComponent},

  {path:"", redirectTo:"emplist",pathMatch:"full"},

  {path:"\*\*", component:PagenotfoundComponent},

];

Step 11: create the service for employee

To consume the data from rest api

11.1 get Employee list

getEmployeeList():Observable<Employee[]>

  {

    return this.http.get<Employee[]>("http://localhost:4040/getEmployeeList")

  }

**11.2 get Employee details**

  getEmployeeDetails(id:number):Observable<Employee>

  {

    let queryparam=new HttpParams().set("id",id);

    return this.http.get<Employee>("http://localhost:4040/getEmployeeDetails",{params:queryparam});

  }

**11.3 Create Employee**

createEmployee(emp:Employee):Observable<Employee>

  {

    return this.http.post<Employee>("http://localhost:4040/createEmployee",emp);

  }

**11.4 update Employee**

updateEmployee(emp:Employee):Observable<Employee>

  {

    return this.http.put<Employee>("http://localhost:4040/updateEmployee",emp);

  }

**11.5 delete Employee**

deleteEmployee(id:Number):Observable<Object>

  {

    return this.http.delete<Object>(`http://localhost:4040/deleteEmployee/${id}`);

  }

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

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

import { Employee } from '../classes/employee';

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class EmployeeService {

  constructor(private http:HttpClient) { }

  getEmployeeList():Observable<Employee[]>

  {

    return this.http.get<Employee[]>("http://localhost:4040/getEmployeeList")

  }

  getEmployeeDetails(id:number):Observable<Employee>

  {

    let queryparam=new HttpParams().set("id",id);

    return this.http.get<Employee>("http://localhost:4040/getEmployeeDetails",{params:queryparam});

  }

  createEmployee(emp:Employee):Observable<Employee>

  {

    return this.http.post<Employee>("http://localhost:4040/createEmployee",emp);

  }

  updateEmployee(emp:Employee):Observable<Employee>

  {

    return this.http.put<Employee>("http://localhost:4040/updateEmployee",emp);

  }

  deleteEmployee(id:Number):Observable<Object>

  {

    return this.http.delete<Object>(`http://localhost:4040/deleteEmployee/${id}`);

  }

}

Step 12: Employee list component details

12 .1 Display List of employees into html

(refer URL for tables ::: <https://getbootstrap.com/docs/4.1/content/tables/>)

<table class="table">

    <thead>

        <tr>

            <th>ID</th>

            <th>firstName</th>

            <th>lastName</th>

            <th>email</th>

            <th>gender</th>

            <th>companyName</th>

            <th>salary</th>

            <th>action</th>

        </tr>

    </thead>

    <tbody>

        <tr \*ngFor ="let emp  of emplist">

            <td>{{emp.id}}</td>

            <td>{{emp.firstName}}</td>

            <td>{{emp.lastName}}</td>

            <td>{{emp.email}}</td>

            <td>{{emp.gender}}</td>

            <td>{{emp.companyName}}</td>

            <td>{{emp.salary}}</td>

            <td>

                <button (click)="updateEmployee(emp.id)" type="button" class="btn btn-primary">Update</button>

                <button  (click)="deleteEmployee(emp.id)" type="button" class="btn btn-danger">Delete</button>

                <button (click)="getEmployee(emp.id)" type="button" class="btn btn-success">GET</button>

            </td>

        </tr>

    </tbody>

</table>

12.2 For list of Employees and update ,delete ,get employee related methods in ts file

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

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

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

import { EmployeeService } from 'src/app/services/employee.service';

@Component({

  selector: 'app-employee-list',

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

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

})

export class EmployeeListComponent {

  public emplist!:any;

  constructor(private empservice:EmployeeService,private router:Router){}

  ngOnInit(): void {

    console.log(this.empservice.getEmployeeList());

    this.empservice.getEmployeeList().subscribe(

      data=>{this.emplist=data}

    )

  }

  updateEmployee(id:number){

    this.router.navigate(['/update',id])

  }

  deleteEmployee(id:number){

    this.empservice.deleteEmployee(id).subscribe(

      data=>{console.log(data)

      this.ngOnInit()}

    )

  }

  getEmployee(id:number ){

    this.router.navigate(['/details',id])

  }

}

Step 13: create Employee data

(refer HTML URL :::: <https://getbootstrap.com/docs/4.0/components/forms/> )

13.1 In html

<form (ngSubmit)="onSubmit()">

    <div class="form-group row">

      <label for="inputFirstNAme" class="col-sm-2 col-form-label">FirstName</label>

      <div class="col-sm-10">

        <input type="text" [(ngModel)]="emp.firstName" name="firstName" class="form-control" id="firstName" placeholder="Enter First Name">

      </div>

    </div>

    <div class="form-group row">

      <label for="inputPassword3" class="col-sm-2 col-form-label">LastName</label>

      <div class="col-sm-10">

        <input type="text" [(ngModel)]="emp.lastName" name="lastName" class="form-control" id="inputPassword3" placeholder="Enter Last Name">

      </div>

    </div>

    <div class="form-group row">

        <label for="inputFirstNAme" class="col-sm-2 col-form-label">Email</label>

        <div class="col-sm-10">

          <input type="email" [(ngModel)]="emp.email" name="email" class="form-control" id="inputEmail3" placeholder="Enter Email">

        </div>

      </div>

      <div class="form-group row">

        <label for="inputPassword3" class="col-sm-2 col-form-label">Gender</label>

        <div class="col-sm-10">

          <input type="text" [(ngModel)]="emp.gender"  name="gender" class="form-control" id="inputPassword3" placeholder="Enter Gender (M/F)">

        </div>

      </div>

      <div class="form-group row">

        <label for="inputPassword3" class="col-sm-2 col-form-label">Company Name</label>

        <div class="col-sm-10">

          <input type="text"  [(ngModel)]="emp.companyName" name="companyName" class="form-control" id="inputPassword3" placeholder="Enter Company Name">

        </div>

      </div>

      <div class="form-group row">

        <label for="inputPassword3" class="col-sm-2 col-form-label">Salary</label>

        <div class="col-sm-10">

          <input type="number" [(ngModel)]="emp.salary" name="salary" class="form-control" id="inputPassword3" placeholder="Enter Salary">

        </div>

      </div>

      <div class="form-group row">

        <div class="col-sm-10">

          <button type="submit" class="btn btn-primary">create Employee</button>

        </div>

      </div>

    </form>

13.2 ts file.

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

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

import { Employee } from 'src/app/classes/employee';

import { EmployeeService } from 'src/app/services/employee.service';

@Component({

  selector: 'app-create-employee',

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

  styleUrls: ['./create-employee.component.css']

})

export class CreateEmployeeComponent {

  emp:Employee=new Employee();

  constructor(private empService:EmployeeService,private router:Router){

  }

  onSubmit()

  {

    this.empService.createEmployee(this.emp).subscribe(

      data=>console.log(data)

    )

    this.router.navigate(['/emplist'])

  }

}

Step 14: Get the Employee details

14.1 html file

    <table class="table">

        <tr>

            <td>

                <label class="form-label"><b>ID</b></label>

            </td>

            <td>

                <label class="form-label">{{empid}}</label>

            </td>

        </tr>

        <tr>

            <td>

                <label   class="form-label"><b>First Name</b></label>

            </td>

            <td>

                <label class="form-label">{{emp.firstName}}</label>

            </td>

        </tr>

        <tr>

            <td>

                <label   class="form-label"><b>Last Name</b></label>

            </td>

            <td>

                <label class="form-label">{{emp.lastName}}</label>

            </td>

        </tr>

        <tr>

            <td>

                <label   class="form-label"><b>Email</b></label>

            </td>

            <td>

                <label class="form-label">{{emp.email}}</label>

            </td>

        </tr>

        <tr>

            <td>

                <label   class="form-label"><b>Gender</b></label>

            </td>

            <td>

                <label class="form-label">{{emp.gender}}</label>

            </td>

        </tr>

        <tr>

            <td>

                <label   class="form-label"><b>Company name</b></label>

            </td>

            <td>

                <label class="form-label">{{emp.companyName}}</label>

            </td>

        </tr>

        <tr>

            <td>

                <label   class="form-label"><b>Salary</b></label>

            </td>

            <td>

                <label class="form-label">{{emp.salary}}</label>

            </td>

        </tr>

    </table>

    <button (click)="gobackHome()" type="button" class="btn btn-primary">Goback Home</button>

14.2 ts file

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

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

import { Employee } from 'src/app/classes/employee';

import { EmployeeService } from 'src/app/services/employee.service';

@Component({

  selector: 'app-employee-details',

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

  styleUrls: ['./employee-details.component.css']

})

export class EmployeeDetailsComponent {

  empid:any;

  emp:Employee=new Employee();

  constructor(private activeRouter:ActivatedRoute,private empservice:EmployeeService,private router:Router){}

  ngOnInit(): void {

   this.empid= this.activeRouter.snapshot.params['id'];

   this.empservice.getEmployeeDetails(this.empid).subscribe(

    data=>this.emp=data

   )

  }

  gobackHome()

  {

    this.router.navigate(['/']);

  }

}

Step 15: update Employee details

15.1 in html page

<form (ngSubmit)="onSubmit()">

    <div class="form-group row">

      <label for="inputFirstNAme" class="col-sm-2 col-form-label">FirstName</label>

      <div class="col-sm-10">

        <input type="text" [(ngModel)]="emp.firstName" name="firstName" class="form-control" id="firstName" placeholder="Enter First Name">

      </div>

    </div>

    <div class="form-group row">

      <label for="inputPassword3" class="col-sm-2 col-form-label">LastName</label>

      <div class="col-sm-10">

        <input type="text" [(ngModel)]="emp.lastName" name="lastName" class="form-control" id="inputPassword3" placeholder="Enter Last Name">

      </div>

    </div>

    <div class="form-group row">

        <label for="inputFirstNAme" class="col-sm-2 col-form-label">Email</label>

        <div class="col-sm-10">

          <input type="email" [(ngModel)]="emp.email" name="email" class="form-control" id="inputEmail3" placeholder="Enter Email">

        </div>

      </div>

      <div class="form-group row">

        <label for="inputPassword3" class="col-sm-2 col-form-label">Gender</label>

        <div class="col-sm-10">

          <input type="text" [(ngModel)]="emp.gender"  name="gender" class="form-control" id="inputPassword3" placeholder="Enter Gender (M/F)">

        </div>

      </div>

      <div class="form-group row">

        <label for="inputPassword3" class="col-sm-2 col-form-label">Company Name</label>

        <div class="col-sm-10">

          <input type="text"  [(ngModel)]="emp.companyName" name="companyName" class="form-control" id="inputPassword3" placeholder="Enter Company Name">

        </div>

      </div>

      <div class="form-group row">

        <label for="inputPassword3" class="col-sm-2 col-form-label">Salary</label>

        <div class="col-sm-10">

          <input type="number" [(ngModel)]="emp.salary" name="salary" class="form-control" id="inputPassword3" placeholder="Enter Salary">

        </div>

      </div>

      <div class="form-group row">

        <div class="col-sm-10">

          <button type="submit" class="btn btn-primary">Update Employee</button>

        </div>

      </div>

    </form>