Custom Service Demo

export interface IEmployee {

code: string;

name: string;

gender: string;

annualSalary: number;

dateOfBirth: string;

}

EmployeeService.ts

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

import { IEmployee } from './IEmployee';

@Injectable()

export class EmployeeService {

getEmployees(): IEmployee[] {

return [

{

code: 'emp101', name: 'Tom', gender: 'Male',

annualSalary: 5500, dateOfBirth: '6/25/1988'

},

{

code: 'emp102', name: 'Alex', gender: 'Male',

annualSalary: 5700.95, dateOfBirth: '9/6/1982'

},

];

}

}

EmployeeComponent.ts

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

import { EmployeeService } from './employee.service';

import { IEmployee } from './IEmployee';

@Component({

selector: 'app-employeeList',

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

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

providers:[EmployeeService]

})

export class EmployeeComponent implements OnInit {

employeerecords: IEmployee[];

// Inject EmployeeService using the constructor

// The private variable \_employeeService which points to

// EmployeeService singelton instance is then available

// throughout this class

constructor(private \_employeeService: EmployeeService) {

}

// In ngOnInit() life cycle hook call the getEmployees()

// service method of EmployeeService using the private

// variable \_employeeService

ngOnInit() {

this.employeerecords = this.\_employeeService.getEmployees();

}

}

Employee.component.html

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

<table>

<thead>

<tr>

<th>Code</th>

<th>Name</th>

<th>Gender</th>

<th>Annual Salary</th>

<th>Date of Birth</th>

</tr>

</thead>

<tbody>

<tr \*ngFor='let employee of employeerecords'>

<!-- <tr \*ngFor='let employee of employeerecords'> -->

<td>{{employee.code}}</td>

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

<td>{{employee.gender}}</td>

<td>{{employee.annualSalary}}</td>

<td>{{employee.dateOfBirth}}</td>

</tr>

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

<td colspan="6">

No Employee Records to Display

</td>

</tr>

</tbody>

</table>

<br />

<!--

<button (click)='getEmployees()'>Refresh Employees</button>

-->

AppModule.ts

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

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

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

import { EmployeeComponent } from './employee/employee.component';

import { HttpRestComponent } from './http-rest/http-rest.component';

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

@NgModule({

declarations: [

AppComponent,

EmployeeComponent,

HttpRestComponent

],

imports: [

BrowserModule,HttpModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

Http Service Demo

REST Service

**package** com.jsonpack;

**import** javax.ws.rs.GET;

**import** javax.ws.rs.Path;

**import** javax.ws.rs.PathParam;

**import** javax.ws.rs.Produces;

**import** javax.ws.rs.core.MediaType;

**import** javax.ws.rs.core.Response;

**import** com.sun.xml.internal.ws.client.ResponseContext;

@Path("/client")

**public** **class** JsonFromRestful {

@GET

@Path("/{clientNo}")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** Response produceCustomerDetailsinJSON(@PathParam("clientNo") **int** no) {

Client client = **new** Client();

client .setClientNo(no);

client .setClientName("John Deer");

**return** Response.*ok*()

.entity(client)

.header("Access-Control-Allow-Origin", "\*")

.header("Access-Control-Allow-Methods", "GET, POST, DELETE, PUT")

.build();

}

@GET

@Path("/clientlist")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** Response produceClientListInJSON() {

Client client1 = **new** Client();

client1 .setClientNo(101);

client1 .setClientName("John Deer");

Client client2 = **new** Client();

client2 .setClientNo(201);

client2 .setClientName("SLB");

Client clist[] = **new** Client[2];

clist[0] = client1;

clist[1] = client2;

**return** Response.*ok*()

.entity(clist)

.header("Access-Control-Allow-Origin", "\*")

.header("Access-Control-Allow-Methods", "GET, POST, DELETE, PUT")

.build();

}

}

IClient.ts

export interface IClient {

clientNo: string;

clientName: string;

}

Client.service.ts

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

import { IClient } from './IClient';

// Import Http & Response from angular HTTP module

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

// Import Observable from rxjs/Observable

import { Observable } from 'rxjs/Observable';

// Import the map operator

import 'rxjs/add/operator/map';

@Injectable()

export class ClientService {

// clientList : IClient

option : RequestOptions

myheaders : Headers

// Inject Angular http service

constructor(private \_http: Http) {

this.myheaders = new Headers();

this.myheaders.append('Access-Control-Allow-Origin', 'http://localhost:4200');

let options = new RequestOptions({headers:this.myheaders})

}

// Notice the method return type is Observable<IClient[]>

getClient(): Observable<IClient[]> {

// To convert Observable<Response> to Observable<IClient[]>

// we are using the map operator

return this.\_http.get('http://localhost:8080/JsonDemo/rest/client/clientlist',this.option)

.map((response: Response) => <IClient[]>response.json());

}

}

HttpRestComponent.ts

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

import { IClient } from './IClient';

import { ClientService } from './client.service';

@Component({

selector: 'app-http-rest',

templateUrl: './http-rest.component.html',

styleUrls: ['./http-rest.component.css'],

providers:[ClientService]

})

export class HttpRestComponent implements OnInit {

clientList: IClient[];

// Inject EmployeeService using the constructor

// The private variable \_employeeService which points to

// EmployeeService singelton instance is then available

// throughout this class

constructor(private \_clientService: ClientService) {

}

// In ngOnInit() life cycle hook call the getEmployees()

// service method of EmployeeService using the private

// variable \_employeeService

// ngOnInit() {

// this.employeerecords = this.\_clientService.getClient();

// }

ngOnInit() {

this.\_clientService.getClient()

.subscribe(clientData => this.clientList = clientData);

}

}

Httprest.component.html

<table>

<thead>

<tr>

<th>ClientCode</th>

<th>Client Name</th>

</tr>

</thead>

<tbody>

<tr \*ngFor='let client of clientList'>

<td>{{client.clientNo}}</td>

<td>{{client.clientName}}</td>

</tr>

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

<td colspan="6">

No Client Records to Display

</td>

</tr>

</tbody>

</table>

<br />