

Comunicación entre componentes

Cuarta Parte

Ph. D. Lenin G. Lemus Zúñiga*

October 7, 2019

Contents

1	Objetivo	1
2	Pasos	1
3	Fichero diary.service.ts	2
4	Deshabilitar control de acceso	3
4.1	header.component.html	3
4.2	fichero app.routing.ts	4
5	diary.component.html	4
6	todo-listing.component.html	8

1 Objetivo

Agregar el acceso a Base de datos

2 Pasos

Los pasos a seguir son los siguientes:

Partiendo de la aplicación con login, se van a modificar los siguientes ficheros:

1. diary.service.ts
2. header.component.html

*lenin.lemus@hotmail.com

3 Fichero diary.service.ts

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpHeaders } from '@angular/common/http';
3 import { Observable } from 'rxjs';
4 import { Todo } from '../models/todo';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class DiaryService {
10   readAllTodosURL = "http://localhost:3100/diary/todos/";
11   readTodoByIDURL = "http://localhost:3100/diary/todo/read";
12   createTodoURL = "http://localhost:3100/diary/todo/create";
13   deleteTodoURL = "http://localhost:3100/diary/todo/delete/";
14
15   httpOptions = {
16     headers: new HttpHeaders({
17       'Content-Type': 'application/json'
18     })
19   };
20
21   constructor(private http: HttpClient) {}
22
23   getTodoList(): Observable<any> {
24     console.log(this.readAllTodosURL);
25     return this.http.get(this.readAllTodosURL, this.httpOptions);
26   }
27   deleteTodoByID(id:string): Observable<any>{
28     return this.http.delete('http://localhost:3100/diary/todo/delete/'+id, this.httpOptions);
29   }
30   createTodo(todo: Todo): Observable<any>{
31     return this.http.post('http://localhost:3100/diary/todo/create', todo , this.httpOptions)
32   }
33   updateTodo(todo):Observable<any>{
34     const miObjeto = JSON.parse(todo);
35     delete(miObjeto['index']);
36     return this.http.patch('http://localhost:3100/diary/'+
```

```

37     miObjeto.id, JSON.stringify(miObjeto) , this.httpOptions);
38 }

```

4 Deshabilitar control de acceso

Se recomienda deshabilitar el control de acceso, para ello edite los ficheros:

- header.component.html
- footer.component.html

4.1 header.component.html

```

1 <mat-toolbar color="primary" > <!--*ngIf="isLoggedIn$ | async
  as isLoggedIn"-->
2   <mat-toolbar-row>
3     <span>
4       
5       Lenin Lemus' Web Site
6     </span>
7     <span class="spacer"></span>
8     <mat-icon routerLink="/principal" class="icon" aria-
hidden="false" aria-label="Home">home</mat-icon>
9     <button mat-button routerLink="/diary">Diary</button>
10    <button mat-button routerLink="/studentManagement">
Students Mgm</button>
11    <button mat-button routerLink="/"
visitedPlaceManagement">Visited Places Mgm</button>
12    <button mat-button routerLink="/about">About</button>
13    <button mat-button (click)="onLogout()">Logout</button>
14  </mat-toolbar-row>
15 </mat-toolbar>

```

subsection footer.component.html

```

1 <mat-toolbar color="primary" > <!--*ngIf="isLoggedIn$ | async
  as isLoggedIn"-->
2   <mat-toolbar-row>
3     <span>&copy; 2019. Lenin Lemus </span>
4     <span class="spacer"></span>
5     <mat-icon class="icon" aria-hidden="false" aria-label
="Example heart icon">favorite</mat-icon>
6   </mat-toolbar-row>
7 </mat-toolbar>

```

4.2 fichero app.routing.ts

```

1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { PrincipalComponent } from '../principal/principal.
  component';
4 import { NotFoundComponent } from '../not-found/not-found.
  component';
5 import { AboutComponent } from '../about/about.component';
6 import { AuthGuard } from '../auth/auth.guard';
7 import { LoginComponent } from '../login/login.component';
8
9 const routes: Routes = [
10   { path: '', redirectTo: 'principal', pathMatch: 'full' },
11     //, canActivate: [AuthGuard]
12   { path: 'login', component: LoginComponent },
13   { path: 'principal', component: PrincipalComponent}, //,
14     canActivate: [AuthGuard]
15   { path: 'about', component: AboutComponent },
16   { path: 'diary', loadChildren: () => import('../diary/diary.
17     module').then(m => m.DairyModule) }, //, canActivate:
18     [AuthGuard]
19   { path: 'studentManagement', loadChildren: '../student/
20     student.module#StudentModule'}, //, canActivate:
21     [AuthGuard]
22   { path: 'visitedPlaceManagement', loadChildren: () => import
23     ('../visited-place/visited-place.module').then(m => m.
24     VisitedPlaceModule) }, //, canActivate: [AuthGuard]
25
26   { path: '**', component: NotFoundComponent},
27 ];
28
29 @NgModule({
30   imports: [RouterModule.forRoot(routes)],
31   exports: [RouterModule]
32 })
33 export class AppRoutingModule { }

```

5 diary.component.html

```

1 import { Component, OnInit } from '@angular/core';
2 import { FormGroup, FormBuilder, Validators } from '@angular/
  forms';
3 import { Todo } from 'src/app/models/todo';
4 import { DiaryService } from 'src/app/services/diary.service';

```

```
5
6 @Component ({
7   selector: 'app-todo-listing',
8   templateUrl: './todo-listing.component.html',
9   styleUrls: ['./todo-listing.component.scss']
10 })
11 export class TodoListingComponent implements OnInit {
12   sidenavActive=true;
13   todoForm: FormGroup;
14   createUpdate=false;
15   selectedTodo: Todo = undefined;
16   selectedIndex:number = undefined;
17
18   DiaryList: Array<Todo> = [
19     { id: '1111', title:'Todo 1', description: 'description
20       Todo 1', isCompleted: false},
21     { id: '2222', title:'Todo 2', description: 'description
22       Todo 2', isCompleted: false},
23     { id: '3333', title:'Todo 3', description: 'description
24       Todo 3', isCompleted: false},
25     { id: '4444', title:'Todo 4', description: 'description
26       Todo 4', isCompleted: false},
27     { id: '5555', title:'Todo 5', description: 'description
28       Todo 5', isCompleted: false},
29     { id: '6666', title:'Todo 6', description: 'description
30       Todo 6', isCompleted: false},
31     { id: '7777', title:'Todo 7', description: 'description
32       Todo 7', isCompleted: false},
33     { id: '8888', title:'Todo 8', description: 'description
34       Todo 8', isCompleted: false},
35     { id: '9999', title:'Todo 9', description: 'description
36       Todo 9', isCompleted: false},
37   ]
38
39   constructor(private fb: FormBuilder, private _dairyService:
40     DiaryService) { // INYECCION DEL SERVICIO
41     this.todoForm = fb.group({
42       title: ['', Validators.required],
43       description: ['', Validators.required],
44       isCompleted: false
45     });
46     this.selectedIndex = undefined;
47     this.selectedTodo = undefined;
48   }
49 }
```

```
40 toggleCreateUpdate() {
41   this.createUpdate = ! this.createUpdate;
42 }
43
44 ngOnInit() {
45   this._dairyService.getTodoList().subscribe((respuesta: any) => { // ACCESO BBDD
46     console.log(respuesta);
47     console.log('datos del servidor');
48     this.DiaryList = respuesta;
49   }, error=>{
50     console.log(error);
51   })
52 }
53
54 setSelectedValues(todo:Todo, indice:number){
55   this.selectedIndex = indice;
56   this.selectedTodo = todo;
57   this.createUpdate = false;
58 }
59
60 processCreateUpdateTodo(todo) {
61   let auxTodo = JSON.parse(todo);
62   if( auxTodo.index === undefined ){
63     console.log('crear');
64     delete(auxTodo['index']);
65     this._dairyService.createTodo(auxTodo).subscribe((
66     respuesta) => { // ACCESO BBDD
67       this.DiaryList.push(respuesta);
68       this.selectedIndex = this.DiaryList.length-1;
69       this.selectedTodo = this.DiaryList[this.selectedIndex
70 ];
71       this.createUpdate = false;
72     })
73   }
74   else{
75     const indice = auxTodo.index;
76     delete(auxTodo.index);
77     this._dairyService.updateTodo(todo).subscribe(() => { //
78     ACCESO BBDD
79       this.DiaryList[indice] = auxTodo;
80       this.selectedTodo=auxTodo;
81     }, (error:any) => {
82       console.log(error);
```

```
81     })
82
83   }
84   this.createUpdate = false;
85 }
86
87 deleteTodo(todo: Todo, index: number) {
88   console.log(todo.id);
89
90   this._dairyService.deleteTodoByID(todo.id).subscribe( ()
91   => { } // ACCESO BBDD
92   this.selectedIndex = undefined;
93   this.selectedTodo = undefined;
94   this.createUpdate = false;
95   this.DiaryList.splice(index, 1);
96 });
97
98 prepareUpdateDirect(todo:Todo, indice:number){
99   this.createUpdate=true;
100   let auxTodo = Object.assign({}, todo);
101   auxTodo.index = indice;
102   this.selectedTodo = auxTodo;
103 }
104
105 prepareUpdate() {
106   this.createUpdate=true;
107   let auxTodo = Object.assign({}, this.selectedTodo);
108   auxTodo.index = this.selectedIndex;
109   console.log(auxTodo);
110   this.selectedTodo = auxTodo;
111 }
112
113 prepareCreate() {
114   this.createUpdate=true;
115   let auxTodo = new Todo();
116   auxTodo.index = undefined;
117   this.selectedTodo = auxTodo;
118 }
119 get date(): string {
120   const fecha = new Date().toLocaleString();
121   console.log(fecha);
122   return fecha;
123 }
124 showNewTodoForm() {
```

```

125     this.selectedTodo = undefined;
126     this.selectedIndex = undefined;
127     this.createUpdate = true;
128   }
129
130
131 }

```

6 todo-listing.component.html

```

1 <ng-container>
2   <mat-toolbar class="header" color="primary">
3     <span>
4       
6       Lenin Lemus' Dairy
7     </span>
8     <button mat-button (click)="sidenav.toggle()">
9       <mat-icon class="icon" aria-hidden="false">dehaze</mat-
10      icon>
11    </button>
12    <span class="spacer"></span>
13    <mat-icon routerLink="/principal" class="icon" aria-hidden
14      ="false" aria-label="Home">home</mat-icon>
15    <button mat-button (click)="showNewTodoForm()">New Todo</
16    button>
17    <button mat-button (click)="sidenav.toggle()">Toggle Show-
18      Hide Sidenav</button>
19    <button mat-button (click)="toggleCreateUpdate()">Toggle
20      Show-Hide CreateUpdate</button>
21  </mat-toolbar>
22
23  <mat-sidenav-container class="container">
24    <mat-sidenav #sidenav mode="side" opened class="sidenav" >
25      <table>
26        <thead><tr><th>Titulo</th><th>Borrar</th></tr></thead>
27        <tbody>
28          <tr *ngFor="let todo of DiaryList; let indice=index
29            ">
30            <td (click)="setSelectedValues(todo, indice)">{{
31              todo.id}}</td> <!--MOSTRAR ID-->
32            <td (click)="setSelectedValues(todo, indice)">{{
33              todo.title}}</td>
34            <td (click)="deleteTodo(todo, indice)"><mat-icon

```



```
26     class="icon" aria-hidden="false">delete</mat-icon> </td>
27     </tr>
28 </tbody>
29 </table>
30 </mat-sidenav>
31
32 <mat-sidenav-content>
33     <h4 *ngIf="!selectedTodo && !createUpdate"> Seleccione
una tarea o pulse en el menu para crear una tarea nueva</h
4>
34     <app-todo-detail *ngIf="selectedTodo && !createUpdate" [
myTodo]="selectedTodo" [myIndex]="selectedIndex" (
eventUpdateTodo)="prepareUpdate(\$event)" ></app-todo-
detail>
35
36     <app-todo-create-update *ngIf="createUpdate" [myTodo]="
selectedTodo" [myIndex]="selectedIndex" (
eventCreateUpdateTodo)="processCreateUpdateTodo(\$event)
"></app-todo-create-update>
37
38 </mat-sidenav-content>
39
40 </mat-sidenav-container>
41
42 <mat-toolbar color="primary" class="footer">Footer</mat-
toolbar>
43 </ng-container>
```