

Formation Angular

TP N°11: Angular Material

SOMMAIRE

1.	Pré-requis :	3
2.	Objectifs	3
3.	Description des services web fournis par la couche Backend	3
4.	Développement de la couche front.....	5
a.	Importer le projet TP11	5
b.	Installer la librairie Material	5
c.	app.module	5
d.	Créer un service.....	6
e.	Modifier le composant Emp-list	7
5.	Test.....	11

1. Pré-requis :

- Réaliser les TPs 1-10 en premier.

2. Objectifs

- ✓ Utiliser la librairie Material.
- ✓ Utiliser le service de pagination fourni par la librairie Material.

3. Description des services web fournis par la couche Backend

- Les services web Rest que nous allons consommer ont été développés avec Spring Boot, Spring Security, JWT et H2.
- Le tableau ci-dessous décrit la liste des SW fournis par la couche Backend :

Service	Le lien	La requête	La réponse
Authentification par le compte admin1	http://localhost:9090/auth/signin Méthode :POST	{ "username":"admin1", ", "password":"admin1", " }	{ "username": "admin1", "jwttoken": "LE TOKEN ICI", "roles": ["ADMIN"] }
Authentification par le compte client1	http://localhost:9090/auth/signin Méthode :POST	{ "username":"client1", "password":"client1", " }	{ "username": "admin1", "jwttoken": "LE TOKEN ICI", "roles": ["CLIENT"] }
Consulter les employés	http://localhost:9090/employees Méthode :GET	Authorization Bearer Token LE TOKEN ICI	Liste des employés : [{ "id": 452, "name": "emp1", "salary": 10000.0, "fonction": "Fonction1" }]
Consulter un employé par son id	http://localhost:9090/employees/7 Méthode :GET	Authorization Bearer Token LE TOKEN ICI	{ "id": 7, "name": "emp1", "salary": 10000.0, "fonction": "Fonction1" }
Ajouter un employé par le profile Admin	http://localhost:9090/admin/create Méthode :POST	Authorization Bearer Token LE TOKEN ICI {	employee is created successfully

		<pre> "name": "nouveau" , "salary": 15000.0 , "fonction": "ING" } </pre>	
Modifier un employé par le profile Admin	http://localhost:9090/admin/update/7 Méthode : PUT	Authorization Bearer Token LE TOKEN ICI <pre> { "name": "nouveau" , "salary": 15000.0 , "fonction": "ING" } </pre>	Employee is updated successfully
Supprimer un employé par le profile Admin	http://localhost:9090/admin/delete/7 Méthode : DELETE	Authorization Bearer Token LE TOKEN ICI	Employee is deleted successfully
Trier les employés par le salaire	http://localhost:9090/employees/sort/salary Méthode : GET	Authorization Bearer Token LE TOKEN ICI	La liste des employés triés par salaire.
Trier les employés par le nom	http://localhost:9090/employees/sort/name Méthode : GET	Authorization Bearer Token LE TOKEN ICI	La liste des employés triés par nom.
Trier les employés par la fonction	http://localhost:9090/employees/sort/fonction Méthode : GET	Authorization Bearer Token LE TOKEN ICI	La liste des employés triés par fonction.
Consulter les employés page par page (service de pagination)	http://localhost:9090/employees/pagination/0/2 Méthode : GET	Authorization Bearer Token LE TOKEN ICI	La liste des employés par page.
Consulter les employés page par page (service de pagination)	http://localhost:9090/employees/paginator?page=0&size=2 Méthode : GET	Authorization Bearer Token LE TOKEN ICI	<pre> { "employees": [{ "id": 7, "name": "emp1", "salary": 10000.0, "fonction": "Fonction1" }, { "id": 8, "name": "emp2", "salary": 20000.0, "fonction": "Fonction3" }], "size": 5 } </pre>

4. Développement de la couche front

a. Importer le projet TP11

- Créer un nouveau dossier (TP11) et copier dedans le contenu du dossier TP10.
- Lancer les commandes suivantes :

- `cd tp11`
- `npm install`

b. Installer la librairie Material

- Lancer la commande : `ng add @angular/material`, ensuite entrer Yes comme le montre l'écran suivant :

```
PS C:\workspace_angular_forGL\tp11> ng add @angular/material
i Using package manager: npm
✓ Found compatible package version: @angular/material@13.3.6.
✓ Package information loaded.

The package @angular/material@13.3.6 will be installed and executed.
Would you like to proceed? Yes
: Installing package...
```

- Choisir ensuite « Indigo/Pink » et cliquer sur Entrer.
- Entrer Yes pour « Set up global Angular Material typography styles »
- Entrer Yes pour « Set up browser animations for Angular Material »

c. app.module

- modifier app.module.ts comme suit :

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { AuthComponent } from './auth/auth.component';
import { NavbarComponent } from './navbar/navbar.component';
import { authInterceptorProviders } from './interceptors/auth.interceptor';
import { WelcomeComponent } from './welcome/welcome.component';
import { EmpListComponent } from './emp/emp-list/emp-list.component';
import { EmpDetailComponent } from './emp/emp-detail/emp-detail.component';
import { EmpCreateComponent } from './emp/emp-create/emp-create.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { MatPaginatorModule } from '@angular/material/paginator';
import { MatProgressSpinnerModule } from '@angular/material/progress-spinner';
```

```

import {MatCardModule} from '@angular/material/card';
import {MatTableModule} from '@angular/material/table';
import {MatButtonModule} from '@angular/material/button';

const materialComponents=[
  MatPaginatorModule,
  MatProgressSpinnerModule,
  MatCardModule,
  MatTableModule,
  MatButtonModule
]

@NgModule({
  declarations: [
    AppComponent,
    AuthComponent,
    NavbarComponent,
    WelcomeComponent,
    EmpListComponent,
    EmpDetailComponent,
    EmpCreateComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    HttpClientModule,
    ReactiveFormsModule,
    BrowserAnimationsModule,
    materialComponents
  ],
  providers: [authInterceptorProviders],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

d. Créer un service

- Modifier la classe emp.service.ts comme suit :

```

import { HttpClient, HttpHeaders, HttpParams } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
const API_GET_EMPLOYEES = 'http://localhost:9090/employees/';
const API_ADMIN_EMPLOYEES = 'http://localhost:9090/admin/';
const httpOptions = {
  headers: new HttpHeaders({ 'Content-Type': 'application/json' })
}

```

```

};

@Injectable({
  providedIn: 'root'
})
export class EmpService {

  constructor(private http: HttpClient) { }

  getAll(): Observable<any> {
    return this.http.get(API_GET_EMPLOYEES);
  }

  getById(id:string): Observable<any> {
    return this.http.get(`${API_GET_EMPLOYEES}/${id}`);
  }

  create(data: any): Observable<any> {
    return this.http.post(API_ADMIN_EMPLOYEES+'create', data,{responseType: 'text'
as 'json'});
  }

  update(id: number, data: any): Observable<string> {
    return this.http.put<string>(`${API_ADMIN_EMPLOYEES+'update'}/${id}`,
data,{responseType: 'text' as 'json'});
  }

  delete(id: number): Observable<string> {
    return this.http.delete<string>(`${API_ADMIN_EMPLOYEES+'delete'}/${id}`,
{responseType: 'text' as 'json'});
  }

  getIntervals(page: string, size: string): Observable<any> {
    let params = new HttpParams();
    params = params.set('page', page);
    params = params.set('size', size);
    return this.http.get<any>(API_GET_EMPLOYEES+'paginator' + '?' +
params.toString())
  }
}

```

e. Modifier le composant Emp-list

- Modifier la classe emp-list.component.ts comme suit :

```

import { Component, OnInit, ViewChild } from '@angular/core';
import { MatPaginator, PageEvent } from '@angular/material/paginator';
import { MatTableDataSource } from '@angular/material/table';
import { Emp } from 'src/app/model/emp';
import { EmpService } from 'src/app/services/emp.service';
import { TokenStorageService } from 'src/app/services/token-storage.service';

@Component({
  selector: 'app-emp-list',
  templateUrl: './emp-list.component.html',
  styleUrls: ['./emp-list.component.css']
})
export class EmpListComponent implements OnInit {

  employees!: any[];
  errorMessage?: string;

  isAdmin: boolean = false;

  loading: boolean = true;
  dataSource = new MatTableDataSource<any>();

  @ViewChild(MatPaginator) paginator!: MatPaginator;
  constructor(private empService: EmpService, private tokenStorageService:
TokenStorageService) {

  }

  ngOnInit(): void {
    this.getData('0', '5')
  }

  pageChanged(event: PageEvent) {
    this.loading = true;
    let pageIndex = event.pageIndex;
    let pageSize = event.pageSize;
    let previousIndex = event.previousPageIndex;
    let previousSize = pageSize * pageIndex;
    this.getNextData(previousSize, (pageIndex).toString(), pageSize.toString());
  }

  getData(page: string, size: string) {

    this.empService.getIntervals(page, size)
      .subscribe((response: any) => {
        if (this.tokenStorageService.hasRole('ADMIN')) {
          this.isAdmin = true;
        }
        this.loading = false;

```



```

        this.employees = response.employees;
        this.employees.length = response.size;
        this.dataSource = new MatTableDataSource<any>(this.employees);
        this.dataSource.paginator = this.paginator;
    },
    err => {
        this.errorMessage = JSON.parse(err.error).message;
    }
)
}

getNextData(currentSize: number, page: string, size: string) {

    this.empService.getInterval(page, size).subscribe((response: any) => {
        this.loading = false;

        this.employees.length = currentSize;
        this.employees.push(...response.employees);

        this.employees.length = response.size;

        this.dataSource = new MatTableDataSource<any>(this.employees);
        this.dataSource._updateChangeSubscription();
        this.dataSource.paginator = this.paginator;

    })
}
}

```

Modifier le template emp-list.component.html comme suit :

```

<div align="center" class="mat-display-1">
  <mat-card>
    <mat-card-header>
      <mat-card-title>
        Liste des employées :
      </mat-card-title>
    </mat-card-header>

    </mat-card>

    <div align="right"> <button [routerLink]="['/',{ outlets: { primary:
['navbar'],contenu: ['create'] } }]" *ngIf="isAdmin" mat-raised-button
color="primary">Nouvel employé</button></div>
    <mat-paginator [pageSizeOptions]="[5, 10, 15]" showFirstLastButtons

```

```

(page)="pageChanged($event)">
</mat-paginator>
<div *ngIf="loading;else table">
  <mat-spinner class="center"></mat-spinner>
</div>

<ng-template #table>
  <table mat-table [dataSource]="dataSource">
    <ng-container matColumnDef="id">
      <th mat-header-cell *matHeaderCellDef> Id </th>
      <td mat-cell *matCellDef="let element"> {{element.id}} </td>

    </ng-container>

    <ng-container matColumnDef="name">
      <th mat-header-cell *matHeaderCellDef> Nom </th>
      <td mat-cell *matCellDef="let element"> {{element.name}} </td>
    </ng-container>

    <ng-container matColumnDef="salary">
      <th mat-header-cell *matHeaderCellDef> Salaire </th>
      <td mat-cell *matCellDef="let element"> {{element.salary}} </td>
    </ng-container>

    <ng-container matColumnDef="fonction">
      <th mat-header-cell *matHeaderCellDef> Fonction </th>
      <td mat-cell *matCellDef="let element"> {{element.fonction}} </td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef=["id', 'name',
'salary', 'fonction']"></tr>
    <tr mat-row *matRowDef="let row; columns: ['id', 'name',
'salary', 'fonction'];"></tr>
  </table>
</ng-template>
</div>

```

Modifier la feuille de style emp-list.component.css comme suit :

```

table {
  width: 100%;
}

```

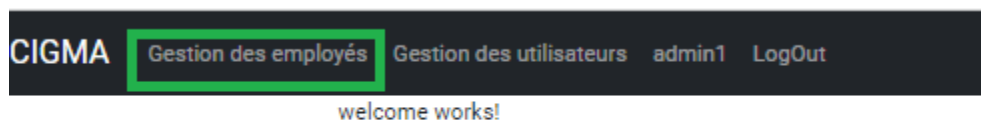
```
.center{
  width: 50%;
  margin: auto;
}
```

5. Test

- Lancer l'application : **ng serve**.
- Accéder au lien <http://localhost:4200/>. La page suivante est affichée :



- Entrer le compte (admin1/admin1) et cliquer sur le bouton « LOG IN ». La page suivante s'affiche :



- Cliquer sur le menu « Gestion des employés ». La page suivante s'affiche :

Liste des employées :

Nouvel employé

Items per page: 5 1 - 5 of 9 |< < > >|

Id	Nom	Salaire	Fonction
7	emp1	10000	Fonction1
8	emp2	20000	Fonction3
9	emp3	30000	Fonction4
10	emp4	40000	Fonction5
11	emp5	50000	Fonction6

- Changer le nombre d'éléments par page en cliquant sur le combo ci-dessous :

Liste des employées :

Nouvel employé

Items per pag 5 1 - 5 of 9 |< < > >|

Id	Nom	Salaire	Fonction
7	emp1	10000	Fonction1
8	emp2	20000	Fonction3
9	emp3	30000	Fonction4
10	emp4	40000	Fonction5
11	emp5	50000	Fonction6

- Naviguer entre les pages en cliquant sur les flèches comme le montre l'écran ci-dessous :

Liste des employées :

Nouvel employé

Items per page: 5 6 - 9 of 9 |< < > >|

Id	Nom	Salaire	Fonction
13	emp7	1500	mm
12	empl6	1000	mm
14	hhh	100	mm
15	lkk	1200	kkk

Vous venez d'utiliser la librairie Material !

Vous pouvez passer au TP n° 12.

Fin du TP 11.