TP5 Angular :
L'authentification et les droits d'accès dans Angular

Objectifs :

- ✓ Créer une page de login,
- ✓ Création du Service Auth,
- ✓ Contextualisation du menu,
- ✓ Création d'un guard,
- ✓ Utilisation de *LocalStorage*.

Créer une page de login

1. Créer le composant Web login :

   ng g c login --skip-tests --inline-style

2. Ajouter le component login au fichier app-routing.module.ts

```
{path: 'login', component: Login},
 {path: '', redirectTo: 'lfilms', pathMatch: 'full' }
```

3. Créer dans le dossier model, la classe User :

```
export class User{
    username:string ;
    password: string ;
    roles:string[];
}
```

4. Ajouter l'attribut user à la classe Login :

```
 user = new User();
```

5. Ajouter la méthode *onLoggedin*() à la classe Login :

```
 onLoggedin()
  {
    console.log(this.user);
  }
```

6. Editer le fichier login.html :

```
imports: [FormsModule],
```

```
<div class="container mt-5">
<div class="row justify-content-md-center">
    <div class="col-md-4">
       <form >
              <div class="form-group">
                   <label >Nom Utilisateur :</label>
                   <input type="text" name="username"
                   class="form-control"
                   [(ngModel)]="user.username" >
               </div>
               <div class="form-group">
                   <label >Mot de passe :</label>
                   <input type="password"    name="password"
                    [(ngModel)]="user.password"    class="form-control">
               </div>
              <div  class="row justify-content-md-center">
                   <button type="button" (click)="onLoggedin()"
                   class="btn btn-success">Connexion</button>
              </div>
         </form>
    </div>
</div>
</div>
```

7. Tester votre travail : http://localhost:4200/login

Création du Service Auth

8. Accéder au dossier services, et y créer le service auth :

   cd .\src\app\services\

   ng g service auth

9. Modifier le fichier auth.service.ts comme suit :

```
users: User[] = [{"username":"admin","password":"123","roles":['ADMIN']},
                 {"username":"kais","password":"123","roles":['USER']}  ];
```

```
  public loggedUser:string;
  public isloggedIn: Boolean = false;
  public roles:string[];



  constructor(private router: Router) { }



logout() {
  this.isloggedIn= false;
  this.loggedUser = undefined;
  this.roles = undefined;
  localStorage.removeItem('loggedUser');
  localStorage.setItem('isloggedIn',String(this.isloggedIn));
  this.router.navigate(['/login']);
}

  SignIn(user :User):Boolean{
    let validUser: Boolean = false;
    this.users.forEach((curUser) => {
      if(user.username=== curUser.username && user.password==curUser.password) {
        validUser = true;
        this.loggedUser = curUser.username;
        this.isloggedIn = true;
        this.roles = curUser.roles;
        localStorage.setItem('loggedUser',this.loggedUser);
        localStorage.setItem('isloggedIn',String(this.isloggedIn));
      }
    });

     return validUser;
  }

  isAdmin():Boolean{
    if (!this.roles) //this.roles== undefiened
       return false;
    return  (this.roles.indexOf('ADMIN') >-1) ;
    ;
  }
```

10. Modifier la méthode *onLoggedin*() de la classe Login :

```
constructor(private authService : AuthService,
            private  router: Router) { }
```

```
onLoggedin(){

 console.log(this.user);
 let isValidUser: Boolean = this.authService.SignIn(this.user);

 if (isValidUser)
    this.router.navigate(['/']);
 else
    alert('Login ou mot de passe incorrecte!');
 }
```

11. Tester votre travail : http://localhost:4200/login
12. Ajouter ces lignes au fichier login. html

```
<div class="row justify-content-md-center">
       <div class="col-md-4">
        <div  class="alert alert-danger" *ngIf="erreur==1">
          <strong>login ou mot de passe erronés..</strong>
        </div>
          <form >
```

13. Ajouter l'attribut erreur à la classe Login:

```
   erreur=0;
```

14. Modifier la méthode onLoggedin() comme suit :

```
else
     //alert('Login ou mot de passe incorrecte!');
     this.erreur = 1;
```

Contextualisation du menu

Afficher l'utilisateur connecté,

15. Modifier la classe App :

```
constructor (public authService: AuthService) {}
```

16. Modifier le fichier app.component.html :

```
<ul class="navbar-nav ml-auto" >
<li> <a  class="nav-link">{{authService.loggedUser}}</a></li>
<li class="nav-item dropdown">
```

Seulement les utilisateurs qui ont le role ADMIN peuvent ajouter des Films.

Donc on va cacher les menu "Ajouter" aux utilisateurs qui ne sont pas des admin.

17. Modifier le fichier app. html :

```
<a *ngIf="authService.isAdmin()"   class="dropdown-item"
routerLink = "/add-Films">Ajouter</a>
```

Seulement les utilisateurs qui ont le role ADMIN peuvent modifier et supprimer des produits

18. Modifier la classe Films :

```
  constructor(private filmtService : FilmService,
              public authService: AuthService) {   }
```

19. Modifier le fichier film,html :

```
<td><a *ngIf="authService.isAdmin()"
class="btn btn-danger" (click)="supprimerfilm(film)">Supprimer</a></td>
<td><a *ngIf="authService.isAdmin()" class="btn btn-
success" [routerLink]="['/updateFilm',film.id]">Modifier</a></td>
```

Cacher ou Montrer les commandes « Login » et « Logout » selon qu'on est connecté ou pas connecté

20. Modifier le fichier app.component.html :

```
<a *ngIf="!authService.isloggedIn" class="dropdown-item" routerLink = "login">login</a>
<a *ngIf="authService.isloggedIn" class="dropdown-item"   (click) = "onLogout()">logout</a>
```

```
onLogout(){
    console.log("logout-------1");
    this.authService.logout();
 }
```

Création d'un guard

Connectez-vous en tant qu'un utilisateur non admin puis essayez d'accéder au formulaire d'ajout de film :

http://localhost:4200/add-Film

est-ce-que vous pouvez le faire ?

21. Créer un guard avec la commande :

    ng g guard Film

    Choisir l'option par défaut (CanActivate)

```
import { inject } from '@angular/core';
import { CanActivateFn, Router } from '@angular/router';
import { Auth } from './services/auth';

export const filmGuard: CanActivateFn = (route, state) => {
const authService = inject(Auth);
  const router = inject(Router);

  if (authService.isAdmin()) {
    return true;
  }

  router.navigate(['app-forbidden']);
  return false

};
```

22. Créer un web component *forbidden* pour afficher le message « Vous n'êtes pas autorisé… »

    ng g c forbidden

23. Modifier le fichier app-routing.module.ts :

```
{path:  'app-forbidden', component: ForbiddenComponent},
```

24. Modifier le fichier forbiden.component.html comme suit :

```
<div class="alert alert-danger" >
    <strong> Vous n'êtes pas autorisé…</strong>
</div>
```

25. Modifier le fichier app-routing.module.ts :

```
{ path: 'add-Films', component: AjoutFilm,canActivate: [filmGuard] },
```

26. Tester : Connectez-vous en tant qu'un utilisateur non admin puis essayez d'accéder au formulaire d'ajout de film :

    http://localhost:4200/add-film

Utilisation de *LocalStorage*

Au démarrage de l'application on teste Localstorage pour voir si l'utilisateur n'est pas déjà connecté dans ce cas on redirige vers login.

27. Modifier la classe App :

```
ngOnInit () {
  let isloggedin: string;
  let loggedUser:string;
  isloggedin = localStorage.getItem('isloggedIn');
  loggedUser = localStorage.getItem('loggedUser');
  if (isloggedin!="true" || !loggedUser)
      this.router.navigate(['/login']);
  else
   this.authService.setLoggedUserFromLocalStorage(loggedUser);
}
```

28. Modifier le fichier auth.service.ts comme suit :

```
setLoggedUserFromLocalStorage(login : string) {
    this.loggedUser = login;
    this.isloggedIn = true;
    this.getUserRoles(login);
  }

  getUserRoles(username :string){
    this.users.forEach((curUser) => {
      if( curUser.username == username ) {
          this.roles = curUser.roles;
      }
    });
  }
```

```
npm install --save concurrently


"scripts": {
  "start": "start": "concurrently \"ng serve --o\" \"json-server --watch
src/db.json\"",

}

npm start :

Angular et JSON Server démarrent en même temps
```

"concurrently \"ng serve --o\" \"json-server --watch src/db.json\"",