

# MIF13 - Programmation web avancée et mobile



# Plan

I. Présentation d'Angular

II. Architecture d'une application Angular

III. Démonstration

# I. Présentation d'Angular

- Framework permettant de créer des applications clientes web en HTML et en TypeScript
- Facilite la création de Single Page Application (SPA)
- Rends l'HTML dynamique

# II. Architecture d'une application Angular

The screenshot displays an IDE with a project explorer on the left and a code editor on the right. The project explorer shows a directory structure for 'exposerangular' with subdirectories 'e2e', 'node\_modules', 'src', and 'assets'. The 'src' directory contains 'app', 'authentication', and 'propose-idea'. The 'app' directory includes 'authentication', 'propose-idea', 'services', and 'show-ideas'. The 'propose-idea' directory contains 'propose-idea.component.html', 'propose-idea.component.scss', 'propose-idea.component.spec.ts', and 'propose-idea.component.ts'. The 'services' directory contains 'app.component.html', 'app.component.scss', 'app.component.spec.ts', and 'app.component.ts'. The 'show-ideas' directory contains 'app.component.html', 'app.component.scss', 'app.component.spec.ts', and 'app.component.ts'. The code editor shows the 'app.module.ts' file with the following code:

```
8 import { CondGuardService } from './services/condGuard.service';
9 import { AuthService } from './services/auth.service';
10 import { IdeaService } from './services/idea.service';
11 import { Routes, RouterModule } from '@angular/router';
12 import { AuthenticationComponent } from './authentication/authentication.component';
13
14 const appRoutes: Routes = [
15   { path: '', component: AuthenticationComponent },
16   { path: 'new', canActivate: [CondGuardService], component: ProposeIdeaComponent },
17   { path: 'show', canActivate: [CondGuardService], component: ShowIdeasComponent },
18   { path: '**', redirectTo: '', pathMatch: 'full' }
19 ];
20
21 @NgModule({
22   declarations: [
23     AppComponent,
24     ProposeIdeaComponent,
25     ShowIdeasComponent,
26     AuthenticationComponent
27   ],
28   imports: [
29     BrowserModule,
30     FormsModule,
31     ReactiveFormsModule,
32     RouterModule.forRoot(appRoutes)
33   ],
34   providers: [
35     AuthService,
36     IdeaService,
37     CondGuardService
38   ],
39   bootstrap: [AppComponent]
40 })
41 export class AppModule { }
```

# Les composants

Composant d'entête

Composant d'authentification

Composant de proposition d'idée

Composant d'affichage des idées

```
app.component.ts
1 import { Component, OnInit } from '@angular/core';
2 import { AuthService } from '../services/auth.service';
3 import { Router } from '@angular/router';
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.scss']
9 })
10
11
12
13 export class AppComponent implements OnInit {
14   connected: boolean;
15   constructor(
16     private authService: AuthService,
17     private router: Router
18   ) {}
19
20   title = 'Student Ideas Proposed';
21   ngOnInit() {
22     this.connected = this.authService.connected();
23   }
24   signOut(){
25     this.authService.deconnexion();
26     this.router.navigate( commands: [ ' ' ] );
27   }
28
29 }
```

# Data Binding

- Insertion dynamique de donnée dans l'application
- One-way data binding injecte du contenu dans la vue : `<h1>{{ name }}</h1>`
- Property binding : `<button [disabled] = « ! isValid »> Valider </button>`
- Two-way data binding : `<input [( name )] = « newName » />`
- Event binding : `<button (click) = 'propose()' </button>`

# Directives structurelles

- Interaction direct avec le DOM de la page HTML
- Ajout – suppression – modification des éléments au cours de l'exécution de la page
- NgIf rend ou non un élément HTML
  - `<div *NgIf= « condition »> Hello World </div>`
- NgFor itère sur une collection afin d'appliquer un template
  - `<li *ngFor = « let idea of ideas » > ... </li>`

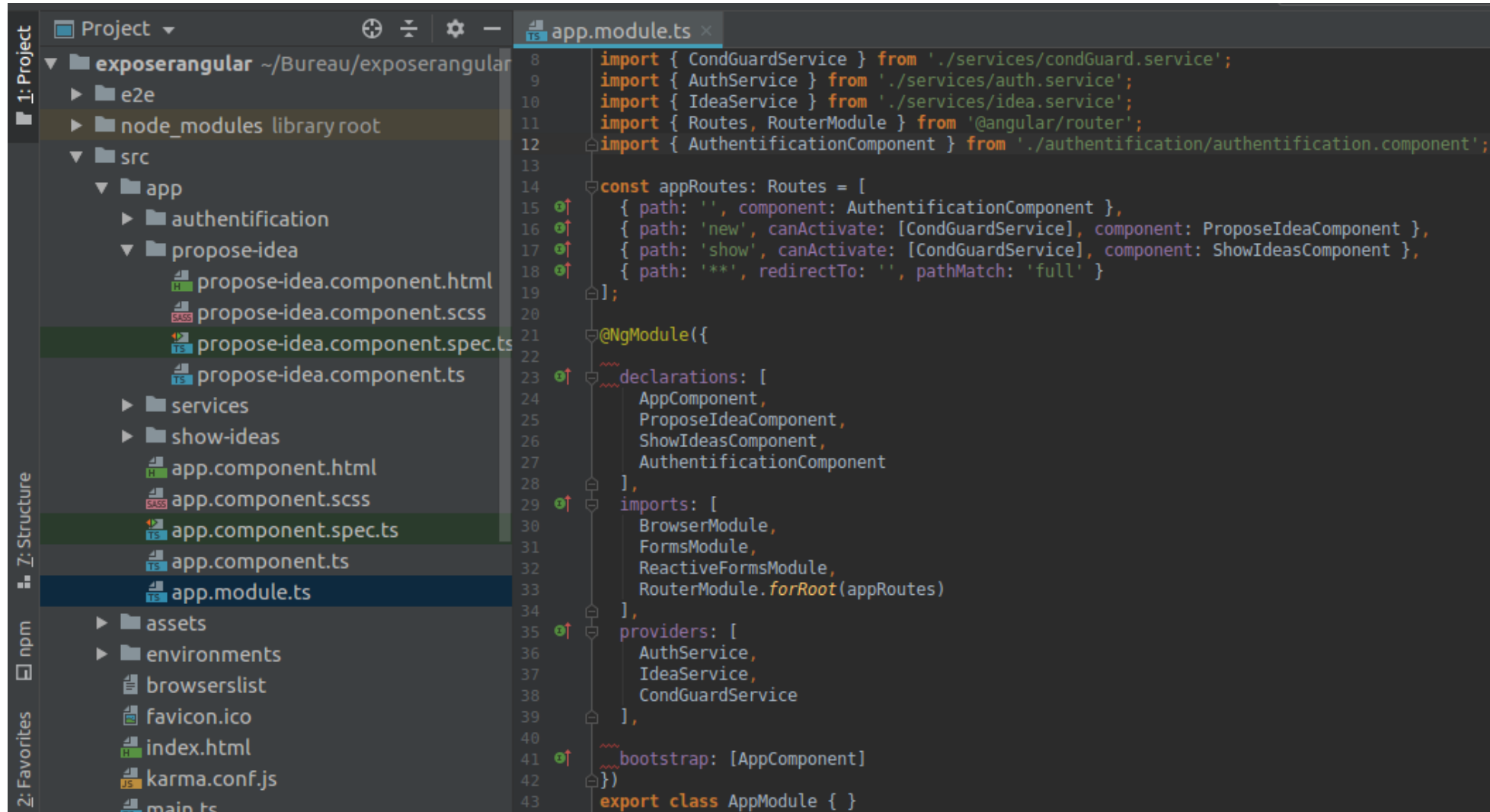
# Services

- Permet de centraliser des parties du code
- Facilite la maintenance, la lisibilité et la stabilité du code

```
auth.service.ts
1  export class AuthService {
2      username = '';
3      password = '';
4      connexion(username, password) {
5          this.username = username;
6          this.password = password;
7      }
8      deconnexion(){
9          this.username = '';
10         this.password = '';
11     }
12     connected(): boolean{
13         return this.username !== '' && this.password !== '';
14     }
15     getUsername(){
16         return this.username;
17     }
18 }
19
```



# Injection des services



The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer displays the following structure:

- Project
- exposerangular ~/Bureau/exposerangular
  - e2e
  - node\_modules libraryroot
  - src
    - app
      - authentication
      - propose-idea
        - propose-idea.component.html
        - propose-idea.component.scss
        - propose-idea.component.spec.ts
        - propose-idea.component.ts
      - services
      - show-ideas
    - app.component.html
    - app.component.scss
    - app.component.spec.ts
    - app.component.ts
    - app.module.ts
  - assets
  - environments
  - browserslist
  - favicon.ico
  - index.html
  - karma.conf.js
  - main.ts

The code editor shows the following TypeScript code for `app.module.ts`:

```
8 import { CondGuardService } from './services/condGuard.service';
9 import { AuthService } from './services/auth.service';
10 import { IdeaService } from './services/idea.service';
11 import { Routes, RouterModule } from '@angular/router';
12 import { AuthenticationComponent } from './authentication/authentication.component';
13
14 const appRoutes: Routes = [
15   { path: '', component: AuthenticationComponent },
16   { path: 'new', canActivate: [CondGuardService], component: ProposeIdeaComponent },
17   { path: 'show', canActivate: [CondGuardService], component: ShowIdeasComponent },
18   { path: '**', redirectTo: '', pathMatch: 'full' }
19 ];
20
21 @NgModule({
22   declarations: [
23     AppComponent,
24     ProposeIdeaComponent,
25     ShowIdeasComponent,
26     AuthenticationComponent
27   ],
28   imports: [
29     BrowserModule,
30     FormsModule,
31     ReactiveFormsModule,
32     RouterModule.forRoot(appRoutes)
33   ],
34   providers: [
35     AuthService,
36     IdeaService,
37     CondGuardService
38   ],
39   bootstrap: [AppComponent]
40 })
41 export class AppModule { }
```

# Routage

```
app.module.ts x
8 import { CondGuardService } from './services/condGuard.service';
9 import { AuthService } from './services/auth.service';
10 import { IdeaService } from './services/idea.service';
11 import { Routes, RouterModule } from '@angular/router';
12 import { AuthenticationComponent } from './authentication/authentication.component';
13
14 const appRoutes: Routes = [
15   { path: '', component: AuthenticationComponent },
16   { path: 'new', canActivate: [CondGuardService], component: ProposeIdeaComponent },
17   { path: 'show', canActivate: [CondGuardService], component: ShowIdeasComponent },
18   { path: '**', redirectTo: '', pathMatch: 'full' }
19 ];
20
```

# III. Démonstration