

Atelier Angular 2 : Routes et Templates Driven Forms

Prérequis

1- Atelier 1

Contenu :

- Navigation
- Services
- Directives : ngFor
- Template Driven Forms

1- Ajout d'un composant ResourceComponent

Nous allons regrouper tous les composants dans un dossier nommé composants :

- Créer le dossier « composants » dans « app »
- Déplacer le dossier « user » dans « composants »
- Mettre à l'import de la classe « User » dans user.component.ts

```
import { User } from '../models/user.models';
```

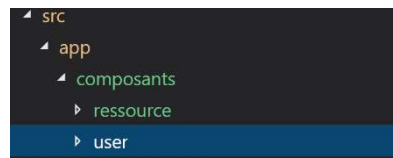
- Mettre à jour l'import dans app.module.ts :

```
import { UserComponent } from './composants/user/user.component';
```

Après ces modifications l'application devra fonctionner comme avant le déplacement du dossier « user »

- Ajouter le composant ResourceComponent dans le dossier composants

```
(src/app/composants)ng g component Resource
```



2- Ajout de la navigation

- Dans le fichier app.module.ts :

- Importer le module RouterModule

```
import { RouterModule } from '@angular/router';
```

- Ajouter RouterModule dans la section 'imports' et configurer les chemins

```
imports: [  
  BrowserModule,  
  RouterModule.forRoot([  
    {path: 'user', component: UserComponent},  
    {path: 'ressource', component: ResourceComponent}])
```

- Remplacer le contenu de « app.component.html » par :

```
<h1>  
  Welcome to {{ title }}!  
</h1>  
<a routerLink="user">Utilisateurs</a>  
<a routerLink="ressource">Ressources</a>  
<router-outlet></router-outlet>
```

- Enlever UserComponent de la collection Bootstrap dans app.module.ts

```
bootstrap: [AppComponent]
```

- d. Supprimer le sélection « <app-user></app-user> » de la page index.html

Le contenu de l'application doit être maintenant comme suit :

Welcome to Angular!

[Utilisateurs](#) [Ressources](#)

User:Angular

Numéro: 5

- 3- Création du service UserService, pour fournir une liste d'utilisateurs
 - a. Créer les dossiers « data » et « services » dans le dossier app
 - b. Dans le dossier « data », ajouter le fichier « user-data.ts », contenant la constante USERS :

```
import { User } from '../models/user.models';

export const USERS: User[] = [
  {numero: 1, nom : 'user1'},
  {numero: 2, nom : 'user2'},
  {numero: 3, nom : 'user3'},
  {numero: 4, nom : 'user4'},
];
```

- c. Créer le service « UserService » dans le dossier « services » :

```
src\app\services>ng g service User
```

- d. Dans « userService », ajouter les imports suivants :

```
import {USERS } from '../data/user-data';
import { User } from '../models/user.models';
```

- e. Ajouter la méthode getUsers dans la classe singleton UserService

```
getUsers(): Promise<User[] > {
  return Promise.resolve(USERS);
}
```

- f. Dans app.module.ts importer « userService » :

```
import { UserService } from './services/user.service';
```

- g. Ajouter UserService dans la section providers de @NgModule

```
providers: [UserService],
```

- 4- Récupération des utilisateurs dans le composant UserComponent
 - a. Ajouter l'attribut users dans le composant UserComponent

```
users: User[] ;
```

- b. Importer UserService
- c. Ajouter une injection de dépendance au niveau du constructeur userService :UserService
- d. Ecrire la méthode getUsers
- e. Appeler la méthode getUsers dans l'événement OnInit

code final de la classe :

```
import { Component, OnInit } from '@angular/core';
import { User } from '../models/user';
import { UserService } from '../services/user.service';
@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
export class UserComponent implements OnInit {
  user: User = { id: 1, nom: 'user1'};
  users: User[];
  // Injection de dépendance
  constructor(private userService: UserService) { }

  getUtilisateurs(): void {this.userService.getUsers ()
    .then(users => this.users = users ); }

  ngOnInit() {
    this.getUtilisateurs();
  }
}
```

5- Remplacer le contenu du fichier template user.component.html, comme suit :

```
<div>
  <h2>Liste des utilisateurs</h2>
<ul>

  <li *ngFor="let u of users">
    {{u.id}} : {{u.nom}}
  </li>
</ul>
```

6- Ajouter un formulaire dans UserComponent

- a. Importer FormsModule dans app.module (nécessaire pour la directive [(ngModel)])

```
import { FormsModule } from '@angular/forms';
imports: [
  BrowserModule,
  FormsModule,
```

- b. ajouter le formulaire suivant, dans le template user.component.html

```
<form >
  Numéro: <input type="text" [value]="user.numero"/>
  Nom: <input type="text" [value]="user.nom"/>
  Nom: <input type="text" name="nom" [(ngModel)]="user.nom"/>
  <button class="btn btn-primary" >Envoyer</button>
</form>
```

c. Dans style.css, importer la feuille de style bootstrap (copier le lien) :

```
@import url(https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css);
```

Projet final : <https://github.com/najasoft/angular-ateliers-2017>