

VI-Routing



GESTION DU ROUTING

AIDE

Cours vidéo (5min):

https://www.youtube.com/watch?v=I2_H6TILVJg&list=PLuWyq_EO5_ALVh9pDGwt7sq6NeXqyaaKv&index=17

L'un des énormes avantages d'utiliser Angular est de pouvoir créer des "single page application" (SPA).

Sur le Web, ces applications sont rapides et lisses : il n'y a qu'un seul chargement de page au début, et même si les données mettent parfois du temps à arriver, la sensation pour l'utilisateur est celle d'une application native.

Au lieu de charger une nouvelle page à chaque clic ou à chaque changement d'URL, on remplace le contenu ou une partie du contenu de la page : on modifie les composants qui y sont affichés, ou le contenu de ces composants.

On accomplit tout cela avec le "routing", où l'application lit le contenu de l'URL pour afficher le ou les composants requis.

I - Les routes

Qu'est-ce qu'une route dans une application Angular ?

Il s'agit des instructions d'affichage à suivre pour chaque URL, c'est-à-dire quel(s) composant(s) il faut afficher à quel(s) endroit(s) pour un URL donné.

Puisque le routing d'une application est fondamentale pour son fonctionnement, on déclare les routes dans `app-routing.module.ts` .

Il est possible d'avoir un fichier séparé pour le routing, mais en termes de fonctionnalité, cela ne change rien : c'est juste une question d'organisation du code.

Modifiez le fichier `app-routing.module.ts` comme ci-dessus

```
code > mon-projet > src > app > TS app-routing.module.ts > ...
1 import { NgModule } from "@angular/core";
2 import { Routes, RouterModule } from "@angular/router";
3 import { MonPremierComponentComponent } from "../mon-premier-component/mon-premier-component.component";
4
5 const routes: Routes = [
6   { path: "presences", component: MonPremierComponentComponent },
7 ];
8
9 @NgModule({
10   imports: [RouterModule.forRoot(routes)],
11   exports: [RouterModule],
12 })
13 export class AppRoutingModule {}
14
```

Le `path` correspond au string qui viendra après le `/` dans l'URL : sur votre serveur local, le premier path ici correspond donc à localhost:4200/presences.

Ne pas ajouter de slash au début de la propriété `path` .

Ensuite, le `composant` correspond au composant que l'on veut afficher lorsque l'utilisateur navigue au `path` choisi.

II - Naviguez avec les routerLink

Afin que l'utilisateur puisse naviguer à l'intérieur de votre application, il est nécessaire de créer des liens ou des boutons qui naviguent vers les routes que vous avez créées.

Comment fait-on ?

On va tout simplement retirer l'attribut `href` et on le remplace par l'attribut `routerLink` auquel on donnera la valeur d'un de nos `path` créé plus tôt.

On appellera aussi `<router-outlet></router-outlet>` afin de faire fonctionner notre code

Modifie le fichier app.component.html par le code suivant

```
code > mon-projet > src > app > < app.component.html > h1
1 | <h1>{{ title }}</h1>
2 | <nav>
3 |   <a class="navbar-brand" routerLink="/">Accueil</a>
4 |   <a class="navbar-brand" routerLink="/presences">Liste</a>
5 | </nav>
6 |
7 | <router-outlet></router-outlet>
8 |
```

Afin de faciliter le test, simplifie le contenu des fichiers de ton composant mon-premier-composant

mon-premier-composant.component.html

```
src > app > mon-premier-composant > < mon-premier-composant.component.html > ...
1 | <h2>Liste de présence</h2>
2 | <li class="list-group-item">
3 |   <p>Etudiant : {{ persoName }} -- Statut : {{ getStatus() }}</p>
4 |   <input type="text" class="form-control" [(ngModel)]="persoName">
5 | </li>
6 |
7 |
```

mon-premier-component.component.ts

```
src > app > mon-premier-component > ts mon-premier-component.component.ts > MonPremierComponentComponent
1 import { Component, Input, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-mon-premier-component',
5   templateUrl: './mon-premier-component.component.html',
6   styleUrls: ['./mon-premier-component.component.css']
7 })
8 export class MonPremierComponentComponent implements OnInit {
9
10   persoName: string = "Emmanuel";
11   persoStatus: string = "Présent";
12
13   constructor() {
14
15   }
16
17   ngOnInit() {
18   }
19
20   getStatus() {
21     return this.persoStatus;
22   }
23 }
```


Résultat:

localhost:4200

[Accueil](#) [Liste](#)

localhost:4200/presences

[Accueil](#) [Liste](#)

Liste des présences

Etudiant : emmanuelle -- Status : Présent

Emmanuelle