

12/12/2022

Version 3

The Angular logo, featuring the word "ANGULAR" in a bold, dark blue, sans-serif font. The text is centered within a white, scalloped-edged circular shape. The background is a solid teal color.

ANGULAR

A decorative wavy line in light blue and white, flowing vertically along the left edge of the slide.

ROUTES

LE ROUTAGE AVEC ANGULAR

ROUTES

- Une seule page HTML, plusieurs points d'accès
 - L'application aura toujours un seul et unique fichier : *index.html*
 - C'est Angular qui va
 - Gérer et interpréter les routes
 - Dispatcher vers le bon composant (un peu à la manière de *Spring MVC*)
 - On utilisera des composants pour jouer le rôle de contrôleurs

ROUTES

- Il est possible de configurer des routes « partielles »
 - Composées d'un ou de plusieurs paramètres
 - « /route/:param/:param2 »
 - « /produit/:id »
 - « /fournisseur/:id/produits »

ROUTES

- Dans le module principal
 - Importer **RouterModule** et **Routes** de `@angular/router`
 - Configuration des routes dans une constante

```
//Configuration des routes
const appRoutes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'produit', component: ProduitComponent },
  { path: 'produit/:id', component: ProduitDetailComponent },
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: '**', component: PageNotFoundComponent }
];
```

Composant « Contrôleur »

pathMatch est important : on veut rediriger seulement si le path est strictement "

- Dans la déclaration des imports de module, importer **RouterModule** en appelant sa méthode `forRoot()`

```
RouterModule.forRoot(appRoutes)
```

ROUTES

- Dans la configuration d'une route
 - `children` Permet d'ajouter des routes secondaires, routes filles
 - `canActivate` Permet de faire jouer un **Guard**

ROUTES

- Le template du composant principal doit implémenter la balise router-outlet

```
<router-outlet></router-outlet>
```

- C'est à cet emplacement qu'iront les vues des « contrôleurs »

ROUTES

- Dans les vues, il est possible de générer les liens, avec ou sans les paramètres

```
<a [routerLink]="['/']">Lien vers l'accueil</a>  
<a [routerLink]="['/produit']">Lien vers les produits</a>  
<a [routerLink]="['/produit', p.id]">Lien vers produit</a>
```

- On peut ajouter une classe « active » lorsque le lien correspond à la page active
 - Il suffit d'utiliser l'attribut « routerLinkActive »

```
<a [routerLink]="['/']" routerLinkActive="active">Lien vers l'accueil</a>  
<a [routerLink]="['/produit']" routerLinkActive="active">Lien vers les produits</a>  
<a [routerLink]="['/produit', p.id]" routerLinkActive="active">Lien vers produit</a>
```


ROUTES

- Dans un contrôleur, il est possible de récupérer les paramètres
 - On a besoin de **ActivatedRoute**, disponible dans *@angular/router*
 - Ensuite, on l'injecte dans le constructeur

```
constructor(private route: ActivatedRoute) { }
```

- Pour lire un paramètre de chemin (PathVariable)

```
this.route.params.subscribe(params => {  
  console.log(params);  
});
```

- Pour lire un paramètre de requête

```
this.route.queryParams.subscribe(params => {  
  console.log(params);  
});
```

ROUTES

- Dans un contrôleur, il est possible de récupérer le **Router** pour rediriger par exemple
 - On a besoin de **Router**, disponible dans *@angular/router*
 - Ensuite, on l'injecte dans le constructeur

```
constructor(private router: Router) { }
```

- Pour rediriger vers une page

```
this.router.navigate([ '/produit', 42 ]);
```

EXERCICE

- Créer un contrôleur **HomeController**
 - N'affiche qu'un HI « Bienvenue »
- Créer un contrôleur **TodoComponent**
 - Affiche la liste des todos
 - Affiche le formulaire d'ajout d'un todo
- Créer un contrôleur **TodoDetailComponent**
 - Attend un paramètre de chemin, avec l'id du todo
 - Affiche les informations du todo, ou une erreur si le todo n'existe pas

ROUTES

- Il est possible d'autoriser les accès à une route selon des conditions
 - Utilisation d'un « Guard »

```
ng generate guard authenticated
```

```
ng g g authenticated
```

EXERCICE

- Créer un CRUD
 - Pouvoir lister les todos
 - En ajouter
 - Les modifier
 - Les supprimer
- Pour aller plus loin (bonus)
 - Pouvoir modifier un produit directement depuis le tableau (des inputs s'affichent dans les cellules)
 - Chaque ligne de todo est un composant **TodoCrudRowComponent**

```
<tr todo-crud-row  
  [todo]="todo"  
  (edited)="onEdit($event)"  
  (deleted)="onDelete($event)">  
</tr>
```