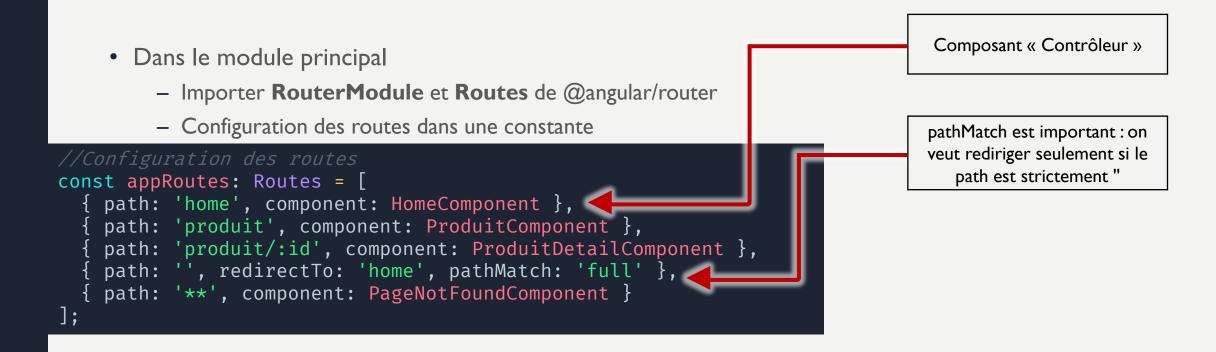


LE ROUTAGE AVEC ANGULAR

- 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

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



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

RouterModule.forRoot(appRoutes)

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

• 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 »

• 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>
```

- 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);
});
```

- 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 HomeComponent
  - N'affiche qu'un H I « 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

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

ng generate guard authenticated

ng 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