

TP2 : Réalisation d'une application Angular

L'objectif du TP est de réaliser une application de vente de Pizza qui affiche toutes les informations concernant les pizzas.

Pizza


marguerite

Mex

4 Fromages

Ajouter une Pizza

Recettes



Mex

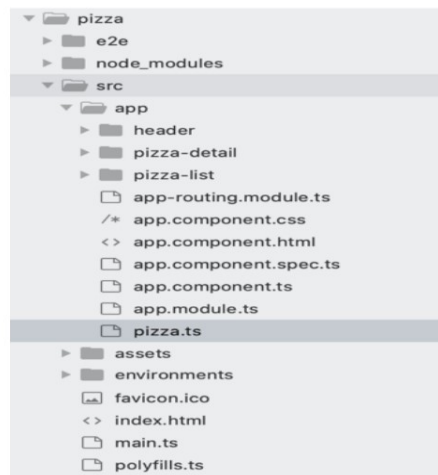
a pizza Tex-Mex, un savant mélange de ...

Editer

1. Création de l'application et liaison de données :

Générez les trois composants suivants :

Pizza-list
pizza-details
header.



- Le modèle Pizza :

Créer un fichier pizza.ts qui contiendra les propriétés de l'objet pizza.

```
export class Pizza {  
  
  public name : string ;  
  public img : string;  
  public desc : string;  
  
  constructor(name: string, img : string, desc: string)  
  {  
    this.name = name;  
    this.img = img;  
    this.desc = desc;  
  }  
}
```

- Dans Pizza list, importer le modèle et déclarer un tableau de Pizza pour afficher la liste des pizzas.
- Dans la template afficher la liste des pizzas dans une balise « a » et en utilisant ngFor.
- Composant detail : Définir une nouvelle Pizza
- Dans la template pizza detail : utilisez l'interpolation pour afficher les différentes propriétés.

2. Input/Output

L'objectif est de réaliser la communication entre les différents composants.

Réorganiser votre application comme ceci :

- Container : gère toute la partie logique
- Detail et list : gèrent la partie affichage
- Pizza.model.ts dans le dossier shared

Output avec (Event emitter) pour émettre la liste au container et Input pour récupérer le détail.
La liste des pizzas est gérée par le container.

1. Création du container
2. Déplacement des données
3. Modification de la template
4. Modifier app.module.ts

La propriété Input()

- Communiquer les pizzas au composant list (binder)
- Importer le décorateur Input dans le container
- Afficher la liste des pizzas et le premier élément du tableau en utilisant le décorateur Input.

La propriété Output()

Afficher la pizza qui correspond au clic. Pour cela utiliser le décorateur Output() pour la communication du composant enfant vers le composant parent. Pour cela :

- Rendre l'index accessible sur la directive *ngFor dans la template list.
- Passer cet index à une méthode recupPizza
- Importer Output et EventEmitter et émettre un événement à chaque fois que la méthode est appelée.
- Récupérer l'événement sur le composant container

3. Directive et services

- Ajouter une directive et un service :
 - o Créer un service Pizza qui gère la liste des pizzas et le détail de la pizza. Ce service servira à partager les données avec la fonctionnalité qui permet de sélectionner une pizza.

- o Créer une directive de style pour modifier le style sur la pizza sélectionnée.
- Dans le dossier shared ajouter un dossier models qui servira à mettre tous les modèles.
Veillez à modifier les chemins des imports vers pizza.model.

Ajouter un nouveau model ingrédient (ingredient.model.ts).

```
export class Ingredient {
  constructor(
    public name: string,
    public quantity: number
  ) {}
}
```

```
import { Ingredient } from '../ingredient.model';
export class Pizza {
  constructor(
    public name: string,
    public img: string,
    public desc: string,
    public ingredients?: Ingredient[] //tu peux instancier sans ingrédient
  ) {}
}
```

Création de la directive :

- Afficher en bleue la pizza sélectionnée
- La directive est créée dans shared/directive

```
labrouk@MacBook-Pro-de-lylia directive % ng g d selection
CREATE src/app/shared/directive/selection.directive.spec.ts (236 bytes)
CREATE src/app/shared/directive/selection.directive.ts (147 bytes)
UPDATE src/app/app.module.ts (924 bytes)
```

Utilisez les décorateurs HostBinding et Input.



Création d'un service qui gère :

- o La liste des pizzas
- o Le détail de la pizza sélectionnée

Créer un dossier services dans shared

- Fournir le service au composant container avec Providers et l'importer.
 - o Dans container ajouter : Providers: ['PizzaService']
 - o import { PizzaService } from '../shared/services/pizza.service';
- Injecter le service dans le composant list et detail
 - o import { PizzaService } from '../shared/services/pizza.service';
 - o constructor(private pizzaService: PizzaListComponent()) { }
- Centraliser les données dans le service :
- Utiliser un BehaviorSubject afin de simuler l'obtention de données asynchrones.
- Utiliser un second BehaviorSubject qui aura la pizza sélectionnée.

```

public pizza : BehaviorSubject<Pizza> = new BehaviorSubject(this.pizzas.value[0])

selectPizza(index: number): void{
  this.pizza.next(this.pizzas.value[index]);
}

constructor() { }
}

```

L'exploiter dans les autres composants

- o Composant détail

```

ngOnInit() {
  this.pizzaService.pizza.subscribe( (pizza: Pizza) => {
    this.pizza = pizza
  })
}

```