

# Compte-rendu Sushi-App

## Table of Contents

Contexte .....	1
Introduction.....	1
Maquette.....	1
Header et Footer .....	4
Affichage des boxes .....	6
Details .....	8
Panier et historique .....	10

Réalisé par :

- Balcerowiak Antoine

site : <https://antoine-balcerowiak.github.io/Sushi-App/menu>

github : <https://github.com/Antoine-Balcerowiak/Sushi-App>

Date début : Mars 2024

## Contexte

2ème projet de seconde année de BTS SIO, le but est de créer un site de vente de boxes de sushi, les boxes de sushi sont récupérées via une API puis affichées sur le site on peut en ajouter dans un panier puis commander et voir l'historique de commande

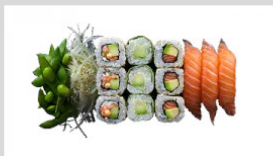
## Introduction

Pour démarrer le projet, j'ai commencé par faire les maquettes de mon site avec Menu, détail, panier, historique et confirmation de la commande

## Maquette

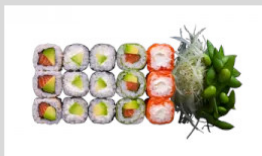


## Boxes sushi :



Tasty blend

12.50 € +



Amateur mix

12.50 € +

[Annuler la commande](#)

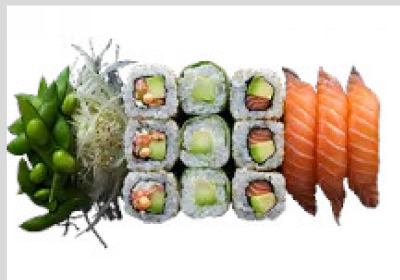
Panier

Validé : 37.50€



## Tasty Blend

## Détails



- x3 California Saumon avocat
- x3 Sushi Saumon
- x3 Spring Avocat Cheese
- x3 California Pacific
- x1 Edamame / Salade de chou

Tasty blend 12.50 € +

[Retourner à la commande](#)[Ajouter au panier](#)



Produit	Quantité	Prix
Tasty blend	— 2 +	25 €
Amateur mix	— 1 +	12.50 €

Annuler

Total : 37.50 €

Payer



## Historique des commandes

Numéro commande	Produit	Date	Prix
567653211	tacos 3 sushi x2 yaourt sushi x1	59/24/-1500	37.50 cailloux
155345514	Salmon Classic x22222 Saumon Original x4	1/1/1939	48.50 franc
878796543	pizza sushi x2 tiramisu sushi x1	42/14/1330	37.50 écus
657564231	Kebab sushi x2 glace sushi x1	37/12/1945	27.50 €

Quitter



## Confirmation de la Commande n° 78548525

## Header et Footer

Tout d'abord j'ai commencer par faire le header et le footer

[...]

```
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
  <li class="nav-item ">
    <a class="nav-link active"
      routerLink="/menu"
      routerLinkActive="active" style="color: white;">
      menu
    </a>
  <li class="nav-item ">
    <a class="nav-link active"
      routerLink="/rgpd"
      routerLinkActive="active" style="color: white;">
      rgpd
    </a>
  </li>
  <li class="nav-item ">
    <a class="nav-link active"
      routerLink="/historique"
      routerLinkActive="active" style="color: white;">
      <i class="bi bi-clock-history"></i> historique
    </a>
  </li>
</ul>
```

[...]

Le Header comporte la page menu qui affiche les boxes sushi que nous allons voir juste après la page RGPD et enfin l'historique que nous allons voir plus tard

Et ensuite le footer

[...]

```
<footer>

  <!-- Button trigger modal -->
  <button type="button" class="btn btn-danger" data-bs-toggle="modal" data-bs-
target="#footer">
    Annuler la commande
  </button>

  <div class="valider">
    <a class="nav-link active" routerLink="/panier" routerLinkActive="active"
style="color: white;">
      Panier : {{ total() | number:'1.2-2'}} €
    </a>
  </div>
</footer>

<!-- Modal -->
<div class="modal fade" id="footer" tabindex="-1" aria-labelledby="exampleModallLabel"
aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModallLabel">Confirmation</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
      <div class="modal-body">
        Êtes-vous sûr de vouloir annuler la commande ?
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="
modal">Annuler</button>
        <button type="button" class="btn btn-danger" (click)="annuler()"
(click)="reload()" > annuler</button>
      </div>
    </div>
  </div>
</div>
```

[...]

Annuler la commande

Panier : 0.00 €

Lui le footer a le bouton panier et annuler commande que nous allons voir après

## Affichage des boxes

Pour l'affichage des boxes tout d'abord il faut les récupérer via l'API

en premier configuration de l'API

source : environment.ts

```
export const environment = {  
  production: true,  
  apiUrl: 'http://localhost:8080/api/boxes',  
};
```

source : environment.developments.ts

```
export const environment = {  
  production: false,  
  apiUrl: 'http://localhost:8080/api/boxes',  
  
};
```

Ensuite dans le service on fait une requête http avec le lien de l'API

source : lookupboxes.service.ts

```
public getBoxes():Observable<any>{  
  return this.http.get(environment.apiUrl)  
}
```

On crée aussi la classe Boxes

source : iBoxes.ts

```
export interface iBoxes {  
  id:number,  
  pieces:number,  
  nom:string,  
  image:string,  
  prix:number,
```

```
saveur:Array<string>,
aliments:Array<Iterable<string>>
}
```

source : Boxes.ts

```
[...]

import { iBoxes } from "./iBoxes";

export class Boxes implements iBoxes {
  constructor(
    public id:number,
    public pieces:number,
    public nom:string,
    public image:string,
    public prix:number,
    public saveur:Array<string>,
    public aliments:Array<Iterable<string>>){}

}
```

Puis maintenant on récupèrent les données des Boxes depuis l'API et en les assignant à la propriété boxes

source: menu.component.ts

```
ngOnInit(): void {
  this.lookupBoxes.getBoxes().subscribe((data)=>{
    this_boxes = data;

  })
}
```

maintenant on les affiche

source: menu.component.html

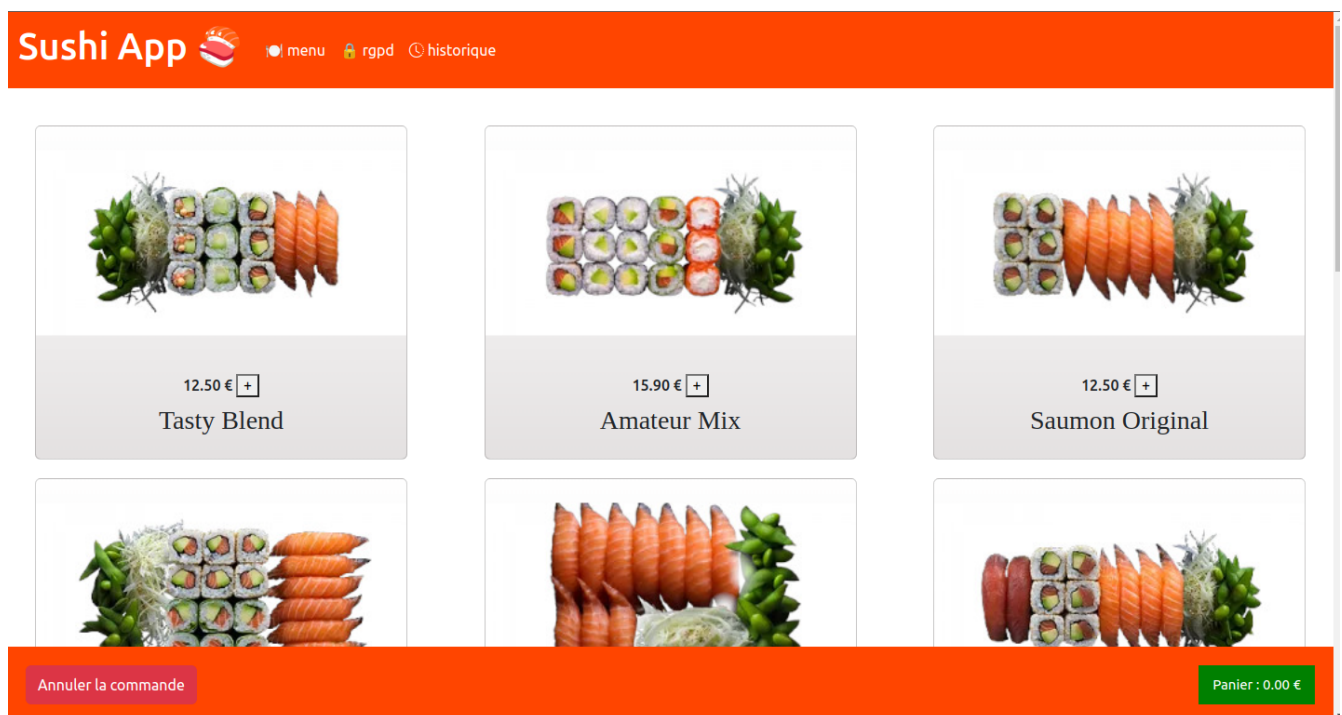
```
<div *ngIf="boxes" class="card-container">
  <div *ngFor="let box of boxes">
    <div class="card">
      <a routerLink="/detail/{{box.id}}" routerLinkActive="active">
        <!-- Image -->
        <div class="card-image"></div>
        <!-- Fin de l'image -->
      </a>
      <!-- Corp de notre carte -->
    </div>
  </div>
</div>
```

```

<div class="card-body">
  <!-- Prix de la box-->
  <div class="card-prix">
    <h6>{{box.prix}}0 € <button
(click)="localStorage(box)">+</button></h6>
  </div>
  <!-- Nom de la box -->
  <div class="card-title">
    <h3>{{box.nom}}</h3>
  </div>
</div>
</div>
</div>
<app-footer></app-footer>

```

Le résultat final



## Details

Pour l'affiche des details ils faut cliquer sur l'image de la boxes

source: menu.component.html

```

<a routerLink="/detail/{{box.id}}" routerLinkActive="active">
  <!-- Image -->
  <div class="card-image"></div>
  <!-- Fin de l'image -->

```



</a>

donc quand on clique sur l'image sa nous emmène sur la page detail avec l'id dans l'url qui correspond a l'id de la boxes

Pour recuperer l'id du lien on utilise

source : detail.component.ts

```
this.route.params.subscribe(params => {  
  this.objetId = params['id'];  
  
});
```

qui surveille les changements des paramètres de l'URL et met à jour objetId avec la valeur du paramètre 'id' à chaque fois qu'il change.

Puis maintenant on récupèrent les données des Boxes depuis l'API et en les assignant à la propriété boxes comme pour menu

source : detail.component.ts

```
ngOnInit(): void {  
  this.lookupBoxes.getBoxes().subscribe((data)=>{  
    this_boxes = data;  
  })  
}
```

Puis enfin dans l'html on affiche les details de la boxes en couplant l'id du lien a tout les id des boxes afin d'afficher les détails de la bonnes boxes

source : detail.component.html

```
<link rel="stylesheet" href="detail.component.css">  
<!-- fichier HTML du composant de détails (detail.component.html) -->  
<div *ngIf="boxes" class="card-container">  
  <div *ngFor="let box of boxes">  
  
    <div class="card" *ngIf="box.id == objetId">  
      <!-- Image -->  
      <div class="card-image">  
          
      </div>  
      <ul>  
        <li><h2 class="nom">{{box.nom}}</h2>  
          <h5>{{box.pieces}} pièces</h5>  
          <h3 class="nom">{{box.prix}}€</h3>  
          <p class="titre">saveurs: </p>  
          <p style="text-transform: capitalize;">{{box.saveurs}}</p>  
        </li>  
      </ul>  
    </div>  
  </div>  
</div>
```

```
<p class="titre">aliments : </p>
<div *ngFor="let aliment of box.aliments">
  <li class="aliments">{{aliment.quantite}} {{aliment.nom}} </li>

</div>
```

Le résultat final

**Sushi App**  [menu](#) [rgpd](#) [historique](#)



### Tasty Blend

12 pièces

**12.50 €**

**saveurs:**

Avocat, Saumon, Cheese

**aliments :**

- 3 California Saumon Avocat
- 3 Sushi Saumon
- 3 Spring Avocat Cheese
- 3 California pacific
- 1 Edamame/Salade de chou

## Panier et historique

tout d'abord pour ajouter au panier il faut cliquer sur le plus qui à coter du prix



12.50 € +

Tasty Blend


Ensuite pour afficher le panier il faut cliquer sur panier dans le footer

Annuler la commande

Panier : 12.50 €

Voila l'affichage du panier avec la box sa quantité son prix total par rapport au nombre de box et le prix total du panier

Sushi App   menu  rgpd  historique

Nom	Quantité	Total	Supprimer
Tasty Blend	- 1 +	12.50 €	
Prix total		12.50 €	

Paye

Annuler la commande

Panier : 12.50 €

puis quand on appuie sur paye sa confirme le paiement

# Confirmation

Commande Validé

ok

puis sa ajoute la commande dans l'historique

Numéro de la commande	Détails de la commande	Prix total de la commande
Commande #1	<div><div>Tasty Blend</div><div>Quantité: 1</div><div>Prix total: 12.50 €</div></div>	12.50 €