

# IMPLEMENTATION DU CALCULATEUR DE RECETTE

## PARTIE 2 : Valorisation recetteDTO et envoi par le service

---

### Objectifs

Dans cette 2e partie nous allons finaliser notre formulaire. Sur la partie 1 nous nous étions concentré sur la récupération des ingrédients disponibles et de leur ajout/suppression de notre recette. Ici nous allons compléter le formulaire avec :

- Le titre de la recette;
- La description;
- Le pourcentage de surgraissage;
- Choix de l'agent alcalin (soude ou potasse);
- Saisir la concentration en agent alcalin en %

### Description du plan d'action :

Pour ce faire, on devra :

- Ajouter une nouvelle classe `LigneIngrédientDTO`
- Modifier la classe `RecetteDTO` pour que le type de `ligneIngrédient` soit un `LigneIngrédientDTO`
- Compléter la logique : ajout des attributs et méthodes
- Compléter la vue : ajout des éléments du formulaire et le bouton de soumission du formulaire

### Finalisation des DTO :

#### Rappel, pourquoi utiliser des DTO pour les objets recette ?

Nous avons défini une classe du côté backend et frontend une classe **Recette**. Cependant notre composant **Angular** ne peut pas soumettre au backend une recette selon cette classe, car elle comporte une liste d'objets de la classe `Resultat`. Or la valorisation (les scores) est réalisé par l'application backend. C'est pour cela que nous envoyons un objet `RecetteDTO` qui contiendra uniquement les attributs utiles au calcul de la recette par le backend.

Dans la même approche, la classe `LigneIngrédient` sera remplacée par `LigneIngrédientDTO`. Ici le problème vient que notre formulaire va nous retourner des objets `Ingrédient` complet, or l'application backend attend uniquement 3 attributs pour les objets `Ingrédient` de `RecetteDTO` :

- L'**ingredientId** (clé primaire de l'ingrédient choisi);
- **pourcentage** pour le pourcentage de l'ingrédient;
- **quantite** pour la masse en g de l'ingrédient.

A FAIRE :

### Création de la classe `LigneIngredientDTO` :

Dans le dossier `models` créer le nouveau fichier : `LigneIngredientDTO.ts`, puis rajouter le contenu suivant :

```
export interface LigneIngredientDTO {
  quantite: number;
  pourcentage: number;
  ingredientId: number;
}
```

### Modification de la classe `RecetteDTO` :

Ouvrir le fichier `RecetteDTO.ts` et modifier le type de `ligneIngredients` pour `LigneIngredientDTO`. Il faudra également importer `LigneIngredientDTO`

Votre classe/interface doit ressembler à cela :

```
import { LigneIngredientDTO } from "../LigneIngredientDTO";

export class RecetteDTO {
  id: number | null = null;
  titre: string = '';
  description: string = '';
  surgraissage: number = 0;
  avecSoude: boolean = false;
  concentrationAlcalin: number = 0;
  qteAlcalin: number = 0;
  ligneIngredients: LigneIngredientDTO[] = [];
}
```

Maintenant nous allons pouvoir compléter notre composant de calcul des recettes.

### Mise à jour de la logique

Ouvrir le fichier **TypeScript** de la logique de votre composant, attention dans le corrigé prof ce composant se nomme `recipe-calculator-page`, mais il peut avoir un nom différent dans votre projet.

Nous allons pouvoir compléter notre classe :

- Ajout d'attributs :
  - `totalMasse`
  - `recetteDto: RecetteDTO`
- Le constructeur va rajouter un attribut `recetteService` (**Attention: UNIQUEMENT si vous avez 2 services séparés pour les recettes et ingrédients !**)
- Ajout des méthodes :
  - `recalculerPourcentages()` : Méthode qui met à jour les pourcentages quand on change les masses des ingrédients.

- `onSubmit()` : Méthode qui va finaliser l'objet `RecetteDTO` et l'envoyer au backend via le service dédié.

### Le code complet de la logique :

```
import { Component, OnInit } from '@angular/core';
import { NgbModal } from '@ng-bootstrap/ng-bootstrap';
import { ModalIngredientPickerComponent } from '../shared/modal-ingredient-picker/modal-ingredient-picker.component';
import { Ingredient } from '../models/Ingredient';
import { IngredientService } from '../services/ingredient.service';
import { LigneIngredient } from '../models/LigneIngredient';
import { LigneIngredientDTO } from '../models/LigneIngredientDTO';
import { RecetteDTO } from '../models/RecetteDTO';
import { Recette } from '../models/Recette';
import { RecetteService } from '../services/recette.service';

@Component({
  selector: 'app-recipe-calculator-page',
  templateUrl: './recipe-calculator-page.component.html',
  styleUrls: ['./recipe-calculator-page.component.css']
})
export class RecipeCalculatorPageComponent implements OnInit {

  // Attributs :
  // -----
  recetteDto: RecetteDTO = new RecetteDTO(); // Objet RecetteDTO de la
nouvelle recette
  availableIngredients: Ingredient[] = []; // A alimenter via service
  selectedIngredients: LigneIngredient[] = []; // Liste des ingrédients
sélectionnés
  totalMasse = 0; // Masse totale des corp gras
  //totalPourcentage = 0;

  // Méthodes :
  // -----

  constructor(
    private ingredientService: IngredientService,
    private recetteService: RecetteService,
    private modalService: NgbModal
  ) {}

  /**
   * Appel du service de récupération des ingrédients à l'initialisation
   */
  ngOnInit(): void {
    this.loadIngredients();
  }
}
```

```
loadIngredients(): void {
  this.ingredientService.getAllIngredients().subscribe({
    next: (ingredients) => {
      this.availableIngredients = ingredients;
    },
    error: (err) => {
      console.error('Erreur lors du chargement des ingrédients', err);
    }
  });
}

/**
 * Modal de sélection des ingrédients.
 */
openIngredientModal(): void {
  const modalRef = this.modalService.open(IngredientPickerComponent);
  modalRef.componentInstance.ingredients = this.availableIngredients;

  modalRef.result.then((selectedIngredient: Ingredient) => {
    if (selectedIngredient) {
      this.ajouterIngredient(selectedIngredient);
    }
  }).catch(() => {});
}

/**
 * Méthode d'ajout d'un ingrédient à la recette
 * @param ingredient Ingrédient à ajouter à la recette
 */
ajouterIngredient(ingredient: Ingredient): void {
  // Empêcher les doublons
  if (this.selectedIngredients.find(l => l.ingredient?.id ===
ingredient.id)) {
    return;
  }

  this.selectedIngredients.push({
    id: 0, // valeur temporaire pour l'instant
    recette: null, // sera renseigné côté backend à la soumission
    ingredient: ingredient,
    quantite: 0,
    pourcentage: 0
  });
}

/**
 * Supprime un ingrédient préalablement choisi pour la recette en cours
 * @param index
 */
supprimerIngredient(index: number): void {
  this.selectedIngredients.splice(index, 1);
}
```

```
/**
 * Recalcule les pourcentages
 */
recalculerPourcentages(): void {
  this.totalMasse = this.selectedIngredients.reduce((acc, ligne) => acc +
    ligne.quantite, 0); // Somme des masse des ingrédients de la recette

  this.selectedIngredients.forEach(ligne => {
    ligne.pourcentage = this.totalMasse > 0 ? + (ligne.quantite /
      this.totalMasse * 100).toFixed(0) : 0; // Calcul les pourcentages des ingrédients
  });
}

/**
 * Méthode de soumission du nouvel ingrédient
 */
onSubmit(): void {
  // Associer les ingrédients au DTO :
  const ligneIngredientDTOs = this.selectedIngredients.map(ligne => ({
    quantite: ligne.quantite,
    pourcentage: ligne.pourcentage,
    ingredientId: ligne.ingredient?.id ?? 0
  }));
  //console.log(`LigneIngredientDTOs = `, ligneIngredientDTOs);

  // Finalisation de l'objet RecetteDTO :
  const recetteEnvoyee: RecetteDTO = {
    ...this.recetteDto,
    ligneIngredients: ligneIngredientDTOs
  };

  //console.log('Objet RecetteDTO prêt à envoyer :', this.recetteDto);

  this.recetteService.addRecetteDTO(recetteEnvoyee).subscribe({
    next: (recette: Recette) => {
      //console.log('Recette reçue du backend :', recette);
      // TODO : afficher résultat, rediriger ou notifier l'utilisateur
    },
    error: (err) => {
      console.error('Erreur lors de la création de la recette :', err);
      // TODO : afficher message d'erreur utilisateur
    }
  });
}
```

## Mise à jour de la vue

Afin que la vue puisse être conforme au rendu attendu, nous allons ajouter :

- Les différents champs de saisie nécessaires;
- Rajouter le bouton de soumission du formulaire avec un appel à la méthode `onSubmit()`

### Le code complet de la vue :

```
<button type="button" class="btn btn-success" (click)="openIngredientModal()">
  + Ingrédient
</button>

<table class="table">
  <thead>
    <tr>
      <th>Nom Ingrédient</th>
      <th>Quantité Gramme</th>
      <th>Pourcentage</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let ligne of selectedIngredients; let i = index">
      <!-- Nom de l'ingrédient -->
      <td>{{ ligne.ingredient?.nom }}</td>

      <!-- Masse de l'ingrédient -->
      <td>
        <input type="number" class="form-control"
          name="quantite{{i}}" [(ngModel)]="ligne.quantite"
          (ngModelChange)="recalculerPourcentages()" min="0">
      </td>

      <!-- Pourcentage de l'ingrédient -->
      <td>
        <input type="number" class="form-control"
          name="pourcentage{{i}}" [(ngModel)]="ligne.pourcentage"
          readonly min="0" max="100">
      </td>

      <!-- Bouton de suppression de l'ingrédient -->
      <td>
        <button type="button" class="btn btn-danger btn-sm"
          (click)="supprimerIngredient(i)">X</button>
      </td>
    </tr>
  </tbody>
</table>
```

```

<div class="mt-3">
  <!-- Masse total des ingrédients -->
  <b>Masse totale :</b> {{ totalMasse | number:'1.0-2' }} g<br>

  <!-- Temporaire la somme des pourcentage (à supprimer) -->
  <!-- <b>Total pourcentage :</b> {{ totalPourcentage | number:'1.0-2' }} % -->
</div>

<!-- Titre -->
<div class="mb-3">
  <label for="titre" class="form-label">Titre :</label>
  <input type="text" id="titre" class="form-control"
    name="titre" [(ngModel)]="recetteDto.titre">
</div>

<!-- Description -->
<div class="mb-3">
  <label for="description" class="form-label">Description :</label>
  <textarea id="description" class="form-control"
    name="description" [(ngModel)]="recetteDto.description"></textarea>
</div>

<!-- Surgraissage -->
<div class="mb-3">
  <label for="surgraissage" class="form-label">Surgraissage (%) :</label>
  <input type="number" id="surgraissage" class="form-control"
    name="surgraissage" [(ngModel)]="recetteDto.surgraissage">
</div>

<!-- Avec Soude -->
<div class="form-check mb-3">
  <input type="checkbox" id="avecSoude" class="form-check-input"
    name="avecSoude" [(ngModel)]="recetteDto.avecSoude">
  <label for="avecSoude" class="form-check-label">Avec Soude</label>
</div>

<!-- Concentration Alcalin -->
<div class="mb-3">
  <label for="concentration" class="form-label">Concentration Alcalin (%) :
</label>
  <input type="number" id="concentration" class="form-control"
    name="concentration" [(ngModel)]="recetteDto.concentrationAlcalin">
</div>

<!-- Bouton soumission de la nouvelle recette -->
<button class="btn btn-primary" (click)="onSubmit()">
  Envoyer
</button>

```