

# IMPLEMENTATION DES GRAPHES EN RADAR

---

## Prérequis

Vous devez avoir votre page "**Mes recettes**" (ou *recettes*; *gestion recettes*; etc.). Si vous avez tenté d'installer dans les séances précédentes **ng2-charts** nous vous invitons **à le désintaller**. En effet il pose de problèmes de compatibilité. Pour le désinstaller, il faudra saisir la commande `npm uninstall chart.js`.

## Installation de Chart.js :

Nous allons utiliser directement **Chart.js** au lieu de passer par l'intermédiaire de **ng2-charts**. Pour l'installation il suffit taper la commande : `npm install chart.js`.

## Création du composant **radar-chart**

### Génération des fichiers avec le CLI d'Angular :

Nous allons intégrer les graphes dans des *cards Bootstrap* et éventuellement le présenter en fin de création d'une recette. Il est donc opportun de créer un composant spécifique que nous pourrons ensuite plus facilement intégrer dans des composants parents.

Si cela n'est pas encore fait, créer votre composant dans le dossier **shared** avec la commande : `ng generate component shared/radar-chart`.

### Implémentation du code de la logique **radar-chart.component.ts** :

Voici le code à placer dans votre fichier :

```
import { Component, AfterViewInit, Input, ViewChild, ElementRef } from
 '@angular/core';
import { Chart, RadarController, RadialLinearScale, PointElement, LineElement,
 Filler, Tooltip, Legend, ChartConfiguration } from 'chart.js';

// 📌 Très important : ce bloc doit être exécuté avant toute création de chart :
Chart.register( RadarController, RadialLinearScale, PointElement, LineElement,
 Filler, Tooltip, Legend);

@Component({
  selector: 'app-radar-chart',
  templateUrl: './radar-chart.component.html',
  styleUrls: ['./radar-chart.component.css'
})
export class RadarChartComponent implements AfterViewInit {
  @Input() title: string = 'Diagramme Radar';
  @Input() scores: number[] = [];

  @ViewChild('radarCanvas') radarCanvas!: ElementRef;
```

```

    readonly labels = [
      'Douceur',
      'Lavant',
      'Volume mousse',
      'Tenue mousse',
      'Dureté',
      'Solubilité',
      'Séchage'
    ];

    ngAfterViewInit(): void {
      new Chart(this.radarCanvas.nativeElement, {
        type: 'radar',
        data: {
          labels: this.labels,
          datasets: [
            {
              label: this.title,
              data: this.scores,
              fill: true,
              backgroundColor: 'rgba(210, 0, 255, 0.2)',
              borderColor: 'rgb(210, 0, 255)',
              pointBackgroundColor: 'rgb(0, 180, 0)',
              pointBorderColor: 'rgb(0, 180, 0)',
              pointHoverBackgroundColor: 'rgb(255, 255, 255)',
              pointHoverBorderColor: 'rgb(0, 180, 0)'
            }
          ]
        },
        options: {
          responsive: true,
          maintainAspectRatio: true,
          scales: {
            r: {
              suggestedMin: 0,
              suggestedMax: 10,
              ticks: { stepSize: 1 }
            }
          },
          plugins: {
            legend: { display: false },
            tooltip: { enabled: true }
          }
        }
      });
    }
  }
}

```

Implémentation du code de la vue `radar-chart.component.html` :

Le code est très simple :

```
<div class="radar-wrapper d-flex justify-content-center">
  <canvas #radarCanvas></canvas>
</div>
```

Implémentation du code CSS pour la vue `radar-chart.component.css` :

Afin que vos graphes soient centrés, un peu de **CSS** :

```
.radar-wrapper {
  width: 100%;
  padding: 1rem;
}

canvas {
  width: 100% !important;
  height: auto !important;
  max-width: 100%;
}
```

## Implémentation dans le composant parent

Il faut maintenant exploiter votre composant graphe dans un composant parent. Afin de vous aider, nous vous fournissons un extrait de composant parents permettant d'afficher toutes les recettes dans des cards :

L'insertion se fait naturellement par la vue du composant parent.

Extrait du code de la vue du parent :

```
<!-- Affichage des cards des recettes -->
<div class="container">
  <div class="row">
    <div *ngFor="let recette of recettes" class="col-12 col-md-6 col-lg-4">
      <div class="card my-2 shadow">

        <!-- Titre de la recette -->
        <div class="card-header">
          <div class="card-title ms-2 my-2">
            <h5><b>{{ recette.titre }}</b></h5>
          </div>
        </div>

        <!-- Graphe radar de la recette : -->
        <div>
          <app-radar-chart
```

```

        [title]="recette.titre"
        [scores]="[
            recette.resultats[2].score,
            recette.resultats[3].score,
            recette.resultats[4].score,
            recette.resultats[5].score,
            recette.resultats[6].score,
            recette.resultats[7].score,
            recette.resultats[8].score
        ]">
    </app-radar-chart>
</div>

<!-- Corps de texte principal de la card -->
<div class="card-body p-2">
    <div class="card-text">
        <div>
            <p class="m-0"><b><u>Description :</u></b></p>
            <p>{{ recette.description }}</p>
        </div>

        <div class="mt-2">
            <b><u>Caractéristiques :</u></b>

            <table>
                <tr>
                    <td><b>- Sugraissage : </b>{{ recette.sugraissage
}}%</td>

                    <td><b>- {{
recette.resultats[0].caracteristique?.nom }} : </b>{{
recette.resultats[0].score.toFixed(1) }}</td>
                </tr>
                <tr>
                    <td><b>- {{
recette.resultats[1].caracteristique?.nom }} : </b>{{
recette.resultats[1].score.toFixed(1) }}</td>
                    <td><b>- {{
recette.resultats[2].caracteristique?.nom }} : </b>{{
recette.resultats[2].score.toFixed(1) }}</td>
                </tr>
                <tr>
                    <td><b>- {{
recette.resultats[3].caracteristique?.nom }} : </b>{{
recette.resultats[3].score.toFixed(1) }}</td>
                    <td><b>- Vol. mousse : </b>{{
recette.resultats[4].score.toFixed(1) }}</td>
                </tr>
                <tr>
                    <td><b>- {{
recette.resultats[5].score.toFixed(1) }}</td>
                    <td><b>- {{
recette.resultats[6].caracteristique?.nom }} : </b>{{
recette.resultats[6].score.toFixed(1) }}</td>
                </tr>
            </table>
        </div>
    </div>

```

```

        <tr>
            <td><b>- {{
recette.resultats[7].caracteristique?.nom }} : </b>{{
recette.resultats[7].score.toFixed(1) }}</td>
            <td><b>- {{
recette.resultats[8].caracteristique?.nom }} : </b>{{
recette.resultats[8].score.toFixed(1) }}</td>
        </tr>
    </table>

</div>

<div class="mt-2">
    <b><u>Composition :</u></b>
    <ul>
        <li><b>Agent alcalin : </b>{{ recette.avecSoude ?
'Soude' : 'Potasse' }}</li>
        <li><b>Quantité alcalin : </b> {{
recette.qteAlcalin.toFixed(0) }}g</li>
        <li><b>Apport eau : </b> {{
recette.apportEnEau.toFixed(0) }}g</li>

        <li class="mt-2"><u><b>Corps gras :</b></u></li>
        <ul>
            <div *ngFor="let ligne of
recette.ligneIngredients">
                <li
*ngIf="ligne.ingredient?.estCorpsGras">
                    {{ ligne.ingredient?.nom }} : {{
ligne.quantite }}g
                </li>
            </div>
        </ul>

        <li class="mt-2"><u><b>Autre :</b></u></li>
        <ul>
            <div *ngFor="let ligne of
recette.ligneIngredients">
                <li *ngIf="ligne.ingredient?.estCorpsGras
== false">
                    {{ ligne.ingredient?.nom }} : {{
ligne.quantite }}g
                </li>
            </div>
        </ul>

    </ul>
</div>
</div>
</div>

<!-- Pied de page de la card -->
<div class="card-footer">
    <a href="#" class="btn btn-info"><i class="bi bi-pen-fill me-1">

```

```
</i>Modifier</a>
      <button href="#" class="btn btn-danger"><i class="bi bi-trash-fill
me-1"></i>Supprimer</button>
    </div>

  </div>
</div>
</div>
</div>
```