



TP5 : LES SERVICES

Matière : Atelier Framework Coté Client

Enseignante : Sonia Guerbouj

Classes : DSI2, MDW2

Durée : 3h

Objectif

Le but de ce TP est d'assimiler le rôle et l'implémentation des services dans une application Angular.

1- Les services

Un service est une classe TypeScript composée d'attributs (propriétés) et de méthodes, dont l'instanciation est gérée par Angular.

Pour faire simple, la responsabilité d'un composant se limite à l'affichage et à la mise en forme des données, toute autre opération doit être déléguée à un service. Les services sont un excellent moyen de partager des informations entre des classes qui ne se connaissent pas.

Utiliser les services a pour but de :

- Réutiliser le code entre différents composants.
- Faciliter l'échange de données.
- Centraliser les appels de service.
- Organiser les rôles : responsabilités visuelles aux composants et fonctionnelles aux services.

Les services sont souvent créés dans les cas suivants :

- Communication avec une API.
- Communication avec une base de données.
- Implémentation d'un cache d'objets.
- Gestion des sessions utilisateur côté client.

La création d'un service se fait à l'aide de la commande :

ng generate service nom_service ou bien simplement : **ng g s nom_service**

Exemple :

```
ng generate service livre
```

2- Travail à faire

Ouvrez l'éditeur de code (VS Code) et le projet « **magasin** » sous le dossier « DSI21 ».

Exercice 1

Dans cet exercice, vous allez modifier le projet « **magasin** » afin d'y intégrer un service.

- 1) Avec VSCode, créez un dossier « **services** » sous le répertoire « **app** » du projet.
- 2) Ouvrez un terminal et créez un service « **article** » sous le dossier « **services** » :

```
C:\Users\USER\Angular\magasin>cd src\app\services  
C:\Users\USER\Angular\magasin\src\app>ng generate service article
```

- 3) Remplacez le contenu du fichier « **article.service.ts** » par :

```
import { Injectable } from '@angular/core';  
import { Article } from '../model/article.model';  
  
@Injectable({  
  providedIn: 'root'  
})  
  
export class ArticleService {  
  tab_art : Article[];  
  
  constructor() {  
    this.tab_art = [  
      {codea : 1, libelle : "Souris Wifi", prix : 39.100, qte : 8,  
dateAjout : new Date("09/27/2022")},  
      {codea : 2, libelle : "Clavier Gaming", prix : 45.900, qte : 11,  
dateAjout : new Date("09/30/2022")},  
      {codea : 3, libelle : "Manette de jeu", prix : 26, qte : 5,  
dateAjout : new Date("10/02/2022")}  
    ];  
  }  
  
  listeArticles():Article[] {  
    return this.tab_art;  
  }  
}
```

- 4) Dans le fichier « **articles.component.ts** », importez le service et modifiez la classe « **ArticlesComponent** » ainsi :

```
import { Component, OnInit } from '@angular/core';  
import { Article } from '../model/article.model';  
import { ArticleService } from '../services/article.service';  
  
@Component({  
  selector: 'app-articles',  
  templateUrl: './articles.component.html',  
  styleUrls: ['./articles.component.css']  
})  
export class ArticlesComponent implements OnInit {  
  tab_art : Article[];  
  
  constructor(private articleService: ArticleService ) {  
    this.tab_art = articleService.listeArticles();  
  }  
}
```

```
ngOnInit(): void {  
  }  
}
```

NB : Dans « **articles.component.htm** », dans la boucle ***ngFor**, remplacez « **articles** » par « **tab_art** ».

5) Exécutez le projet : `ng serve -open` et vérifiez que la liste des articles s'affiche.

Exercice 2

Dans le projet **magasin** de l'exercice 1, on modifie les fichiers du composant **add-article**.

1) Dans le fichier « **add-article.component.ts** », importez les modules de la classe modèle « **Article** » et du service « **article** ». Puis, ajoutez l'attribut « **newArticle** » à la classe « **AddArticleComponent** » ainsi que la méthode « **addArticle** » :

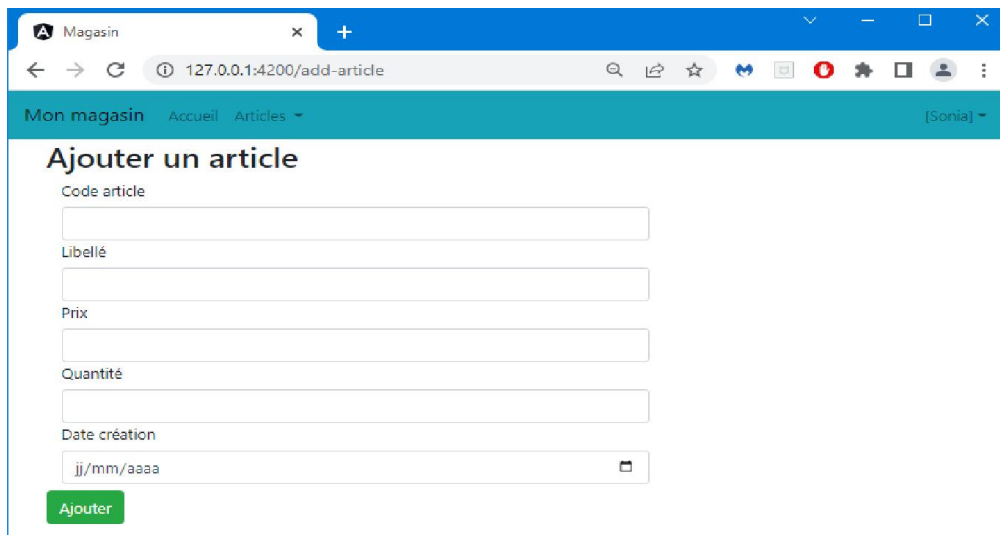
```
import { Component, OnInit } from '@angular/core';  
import { Article } from '../model/article.model';  
import { ArticleService } from '../services/article.service';  
  
@Component({  
  selector: 'app-add-article',  
  templateUrl: './add-article.component.html',  
  styleUrls: ['./add-article.component.css']  
})  
export class AddArticleComponent implements OnInit {  
  newArticle = new Article();  
  
  constructor(private articleService: ArticleService ) { }  
  
  addArticle() {  
    this.articleService.ajouterArticle(this.newArticle);  
  }  
  
  ngOnInit(): void {  
  }  
}
```

2) Avant de compléter la vue (HTML) du composant « **add-article** », il faut ajouter le module « **FormsModule** » dans le fichier « **app.module.ts** » afin de permettre la liaison de données "Two ways data binding" :

```
...  
import { FormsModule } from '@angular/forms';  
...  
@NgModule({  
  ...  
  imports: [  
    BrowserModule,  
    AppRoutingModule, FormsModule  
  ],  
  ...  
})
```

3) Remplacez le contenu du fichier « **add-article.component.html** » comme suit :

```
<div class="container">
  <div>
    <h2>Ajouter un article</h2>
  </div>
  <form >
    <div class="col-sm-2 col-md-2 col-lg-2">
      <label >Code article</label>
      <input type="number" [(ngModel)]="newArticle.codea"
name="code" class="form-control">
    </div>
    <div class="col-sm-4 col-md-4 col-lg-4" >
      <label >Libellé</label>
      <input type="text" [(ngModel)]="newArticle.libelle"
name="libelle" class="form-control">
    </div>
    <div class="col-sm-2 col-md-2 col-lg-2">
      <label >Prix</label>
      <input type="number" [(ngModel)]="newArticle.prix"
name="prix" class="form-control">
    </div>
    <div class="col-sm-2 col-md-2 col-lg-2">
      <label >Quantité</label>
      <input type="number" [(ngModel)]="newArticle.qte"
name="qte" class="form-control">
    </div>
    <div class="col-sm-4 col-md-4 col-lg-4">
      <label >Date création</label>
      <input type="date" [(ngModel)]="newArticle.dateAjout"
name="dateAjout" class="form-control">
    </div>
    <div class="mt-2">
      <button type="submit" (click)="addArticle()" class="btn
btn-success">Ajouter</button>
    </div>
  </form>
</div>
```



Mon magasin Accueil Articles [Sonia]

Ajouter un article

Code article

Libellé

Prix

Quantité

Date création

- 4) Dans le fichier « **article.service.ts** », ajoutez la méthode « ajouterArticle » dans la classe « ArticleService » :

```
ajouterArticle(art: Article) {
    this.tab_art.push(art);
    console.log("Ajout avec succes : "+art.libelle);
}
```

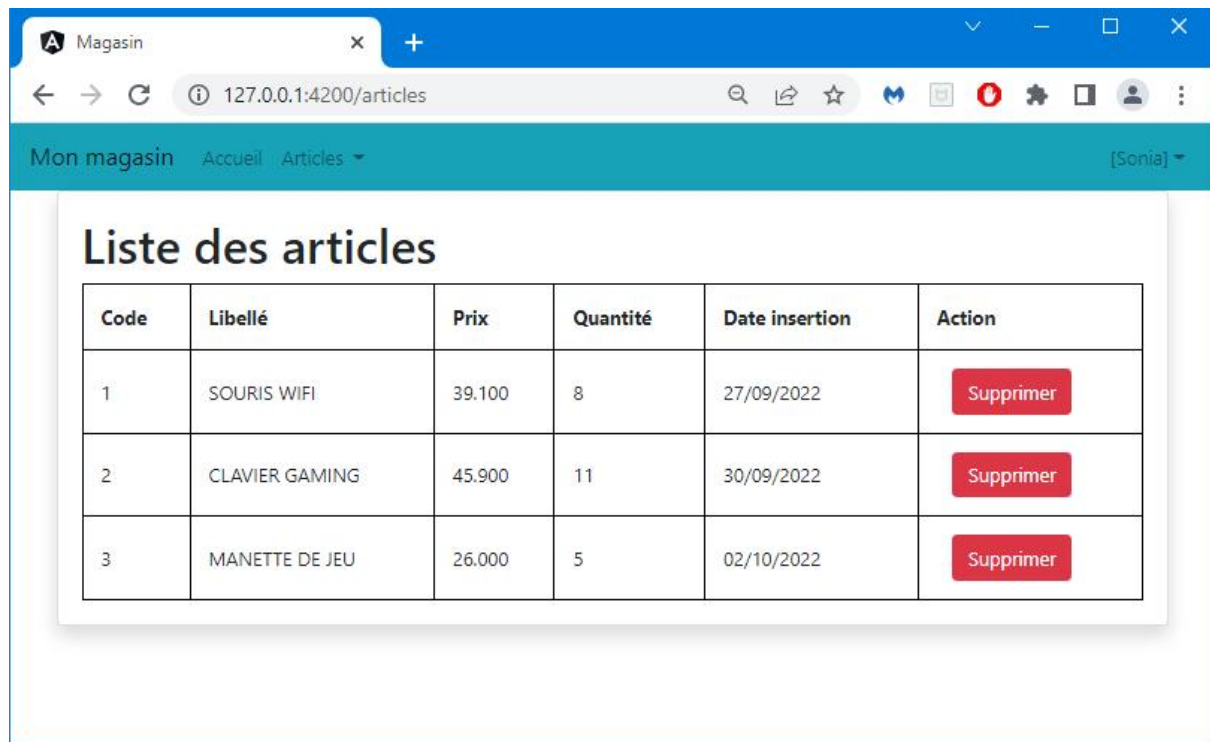
- 5) Dans le navigateur Web, cliquez sur « Articles > Ajouter » dans le menu.
- Remplissez le formulaire d'ajout et validez. Puis, cliquez avec le bouton droit de la souris sur la page web, choisissez « Inspecter », Enfin, cliquez sur l'onglet « Console » pour vérifier l'ajout du nouvel article.
 - Pour une meilleure vérification, dans le fichier « **add-article.component.html** », ajoutez une <div> dans laquelle insérez la balise du composant « articles ». Enregistrez et testez de nouveau le formulaire d'ajout.

Exercice 3

Dans le projet **magasin** de l'exercice 2, on implémente la suppression d'articles.

- 1) Dans le fichier « **articles.component.html** », ajoutez dans le tableau une colonne intitulée « Action », dans laquelle insérez ce code :

```
<td>
    <a class="btn btn-danger" (click)="suppArticle(article)">Supprimer</a>
</td>
```



- 2) Dans le fichier « **article.service.ts** », ajoutez la méthode « **supprimerArticle** » dans la classe « **ArticleService** » afin de supprimer, du tableau « **tab_art** », un article passé en paramètre.

```
supprimerArticle(art: Article) {  
    const index = this.tab_art.indexOf(art, 0);  
    if (index > -1) {  
        this.tab_art.splice(index, 1);  
    }  
}
```

- 3) Dans le fichier « **articles.component.ts** », ajoutez la méthode « **suppArticle** » dans la classe « **ArticlesComponent** ».

```
suppArticle(art: Article) {  
    this.articleService.supprimerArticle(art);  
    console.log("Suppression avec succes :"+art.libelle);  
}
```

- 4) Dans le navigateur, testez l'un des boutons « **supprimer** ».
- 5) On remarque que les articles sont supprimés directement. Pour demander une confirmation avant la suppression, modifiez la méthode « **suppArticle** » dans le fichier « **articles.component.ts** » comme suit puis testez de nouveau :

```
suppArticle(art: Article){  
    let conf = confirm("Etes-vous sûr de vouloir supprimer cet article ?");  
    if (conf) {  
        this.articleService.supprimerArticle(art);  
        console.log("Suppression avec succes :"+art.libelle);  
    }  
}
```