

# **Formation Angular 11**

---

## **TP N°5 : L'utilisation des services dans Angular**

---

# SOMMAIRE

1.	Pré-requis : .....	3
2.	Objectifs .....	3
3.	Développement.....	3
a.	Création de l'application tpsservices .....	3
b.	Création du composant e-info.....	4
c.	Ajouter le Module FormsModule dans app.module.ts .....	4
d.	Création de la classe EmployeeVo.....	5
e.	Création du service data .....	6
f.	Injecter le service.....	7
g.	Modification du Template .....	8
h.	Modification de la feuille de style e-info.component.css .....	9
i.	Modification de la page app.component.html.....	10

### 1. Pré-requis :

- Réaliser les TPs 1-4 en premier.

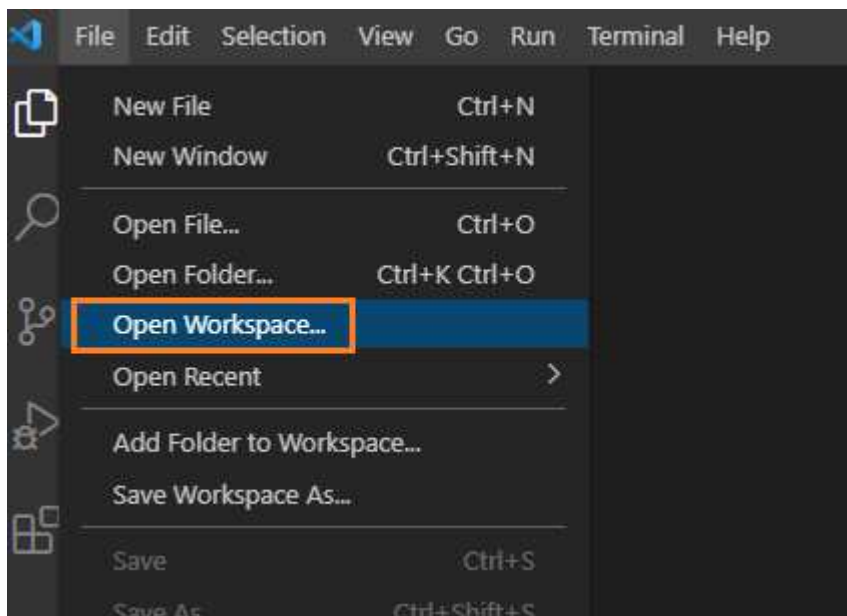
### 2. Objectifs

- ✓ Créer une classe.
- ✓ Injecter un service dans un composant.
- ✓ Utiliser ***l'event binding*** pour implémenter le clic sur un bouton.

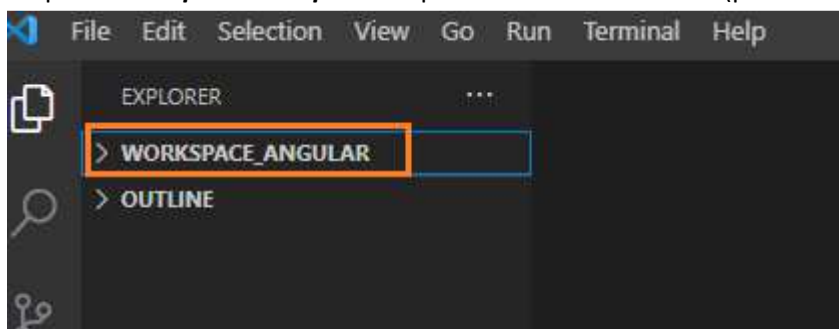
### 3. Développement

#### a. Création de l'application **tpservices**

- Lancer VSC et ouvrir votre workspace en suivant les étapes suivantes :



- Cliquer sur « **Open Workspace...** » puis choisir votre dossier (par exemple c:/workspace\_angular) :



- Ouvrir ensuite un terminal et taper la commande suivante pour créer le projet **tpservices** :

```

PS C:\workspace_Angular> ng n tpservices
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE tpservices/angular.json (3655 bytes)
CREATE tpservices/package.json (1200 bytes)
CREATE tpservices/README.md (1019 bytes)
CREATE tpservices/tsconfig.json (783 bytes)
CREATE tpservices/tslint.json (3185 bytes)

```

- L'application *tpservices* est ensuite générée par Angular CLI.

- Démarrer l'application avec la commande **ng serve**.

#### b. Création du composant e-info

- Dans le terminal, lancer les commandes suivantes :

```

PS C:\workspace_Angular\tpformulaires> cd ../tpservices\
PS C:\workspace_Angular\tpservices>
PS C:\workspace_Angular\tpservices>
PS C:\workspace_Angular\tpservices> ng g c e-info
CREATE src/app/e-info/e-info.component.html (21 bytes)
CREATE src/app/e-info/e-info.component.spec.ts (620 bytes)
CREATE src/app/e-info/e-info.component.ts (274 bytes)
CREATE src/app/e-info/e-info.component.css (0 bytes)
UPDATE src/app/app.module.ts (473 bytes)
PS C:\workspace_Angular\tpservices>

```

#### c. Ajouter le Module FormsModule dans app.module.ts

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { EInfoComponent } from './e-info/e-info.component';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent,
    EInfoComponent
  ],
  imports: [
    BrowserModule,

```

```

    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

#### d. Création de la classe EmployeeVo

- Lancer la commande suivante :

```

PS C:\workspace_Angular\tpservices> ng g class EmployeeVo
CREATE src/app/employee-vo.spec.ts (1/1 bytes)
CREATE src/app/employee-vo.ts (28 bytes)
PS C:\workspace_Angular\tpservices>

```

- Les deux fichiers suivants sont créés :

```

v tpservices
  > e2e
  > node_modules
  v src
    v app
      > e-info
      TS app-routing.module.ts
      # app.component.css
      <> app.component.html
      TS app.component.spec.ts
      TS app.component.ts
      TS app.module.ts
      TS employee-vo.spec.ts
      TS employee-vo.ts
    > assets
    > environments

```

- Modifier la classe EmployeeVo comme suit :

```

export class EmployeeVo {
  id: number;
  firstname: string;
  lastname: string;

  constructor(id: number, firstname: string, lastname: string) {
    this.id = id;
    this.firstname = firstname;
  }
}

```

```

        this.lastname = lastname;
    }
    getId(): number {
        return this.id;
    }

    getLastName(): string {
        return this.lastname;
    }

    getFirstname(): string {
        return this.firstname;
    }

    setId(id:number) {
        this.id=id;
    }

    setFirstname(firstname:string) {
        this.firstname=firstname;
    }

    setLastName(lastname:string) {
        this.lastname=lastname;
    }
}

```

#### e. Création du service data

- Lancer la commande suivante :

```

PS C:\workspace_Angular\tpservices> ng g s data
CREATE src/app/data.service.spec.ts (347 bytes)
CREATE src/app/data.service.ts (133 bytes)
PS C:\workspace_Angular\tpservices>

```

Modifier le fichier **data.service.ts** comme suit :

```

import { Injectable } from '@angular/core';
import { EmployeeVo } from './employee-vo';

@Injectable({
    providedIn: 'root'
})

```

```

export class DataService {
  employees:EmployeeVo[]=
    new EmployeeVo(1,'Alami','Mohammed'),
    new EmployeeVo(2,'Jhon','Jack'),
    new EmployeeVo(3,'Ben','Michel'),
    new EmployeeVo(4,'Alex','Bela'),
    new EmployeeVo(5,'Faty','Younes'),
    new EmployeeVo(6,'Hika','Sami')
];

  constructor() { }

  getData():EmployeeVo[] {
    return this.employees;
  }

  add(employee:EmployeeVo):EmployeeVo[] {
    this.employees.push(employee);
    return this.employees;
  }
}

```

#### f. Injecter le service

- Modifier la classe **e-info.component.ts** comme suit :

```

import { Component, OnInit } from '@angular/core';
import { EmployeeVo } from '../employee-vo';
import { DataService } from '../data.service';

@Component({
  selector: 'info',
  templateUrl: './e-info.component.html',
  styleUrls: ['./e-info.component.css'],
  providers: [DataService]
})
export class EInfoComponent implements OnInit {
  employees: EmployeeVo[] = [];

  constructor(private service: DataService) {
    this.employees = this.service.getData();
  }

  add(frm: any): EmployeeVo[] {

    console.log("id=", frm.value.firstname);

```

```

    this.employees = this.service.add(new EmployeeVo(frm.value.id, frm.value.firstnam
e, frm.value.lastname));
    return this.employees;
  }

  getData(): EmployeeVo[] {
    return this.service.getData();
  }
  ngOnInit(): void {
  }
}

```

#### g. Modification du Template

```

<h1>Inscription</h1>
<div class="container">
  <form #frm="ngForm" (ngSubmit)="add(frm)">

    <div class="form-group">
      <label for="id">Id</label><br>
      <input #id="ngModel" required type="text" name="id" class="form-
control" ngModel>
      <div *ngIf="id.touched && id.invalid" class="alert alert-danger">
        <p *ngIf="id.errors != null && id.errors.required"> id ne peut pas êt
re vide</p>
      </div>
    </div>

    <pre></pre>

    <div class="form-group">
      <label for="firstname">Firstname</label><br>
      <input #firstname="ngModel" required type="text" name="firstname" class="
form-control" ngModel>
      <div *ngIf="firstname.touched && firstname.invalid" class="alert alert-
danger">
        <p *ngIf="firstname.errors != null && firstname.errors.required"> fir
stname ne peut pas être vide</p>
      </div>
    </div>

    <pre></pre>

    <div class="form-group">
      <label for="lastname">Firstname</label><br>
      <input #lastname="ngModel" required type="text" name="lastname" class="fo

```



```

rm-control" ngModel>
    <div *ngIf="lastname.touched && lastname.invalid" class="alert alert-
danger">
        <p *ngIf="lastname.errors != null && lastname.errors.required"> lastn
ame ne peut pas être vide</p>
    </div>
</div>
<pre></pre>
<button class="btn btn-primary" type="submit">submit</button>
</form>
<pre></pre>
<br>
<h3>Liste des employés :</h3>
<ul class="list-group">
    <li *ngFor="let info of employees" class="list-group-item">
        {{info.id}}, {{info.firstname}},{{info.lastname}}
    </li>
</ul>
</div>

```

#### h. Modification de la feuille de style e-info.component.css

```

.container {
    text-align: center;
}
label {
    text-align: left;
}
h1 {
    text-align: center;
}

.alert{
    color: white;
    margin-bottom: 15px;
    text-align: left;
}

.alert-danger {
    background-color: red;
}

.list-group-item{
    text-align: left;
}

```

```
}
```

i. Modification de la page app.component.html

```
<info></info>
```

Lancer ensuite l'URL : <http://localhost:4200> :

## Inscription

Id
<input type="text"/>
Firstname
<input type="text"/>
Firstname
<input type="text"/>
<input type="button" value="submit"/>

### Liste des employés :

- 1, Alami, Mohammed
- 2, Jhon, Jack
- 3, Ben, Michel
- 4, Alex, Bela
- 5, Faty, Younes
- 6, Hika, Sami

Insérer des données dans le formulaire et cliquer sur le bouton submit :

# Inscription

Id
<input type="text" value="12"/>
Firstname
<input type="text" value="BAROO"/>
Firstname
<input type="text" value="Jack"/>
<input type="button" value="submit"/>

## Liste des employés :

- 1, Alami,Mohammed
- 2, Jhon,Jack
- 3, Ben,Michel
- 4, Alex,Bela
- 5, Faty,Younes
- 6, Hika,Sami
- 12, BAROO,Jack

**Vous venez créer un service et l'injecter dans un composant.**

**Maintenant, vous pouvez passer au TP n°6.**

**Fin du TP 5.**