

Formation Angular 11

TP N°4: Gestion des formulaires dans Angular

SOMMAIRE

1.	Pré-requis :	3
2.	Objectifs	3
3.	Développement	3
a.	Création de l'application tpformualires	3
b.	Création du composant Inscription	4
c.	Ajouter le Module FormsModule dans app.module.ts	5
d.	Modification de la page inscription.component.html	5
e.	Modification de la feuille de style inscription.component.html	6
f.	Modification de la page app.component.html	6
g.	La directive ngModel	7
h.	Présentation JavaScript du formulaire	10
i.	Ajouter des règles de validation	12

1. Pré-requis :

- Réaliser les TP 1-3 en premier.

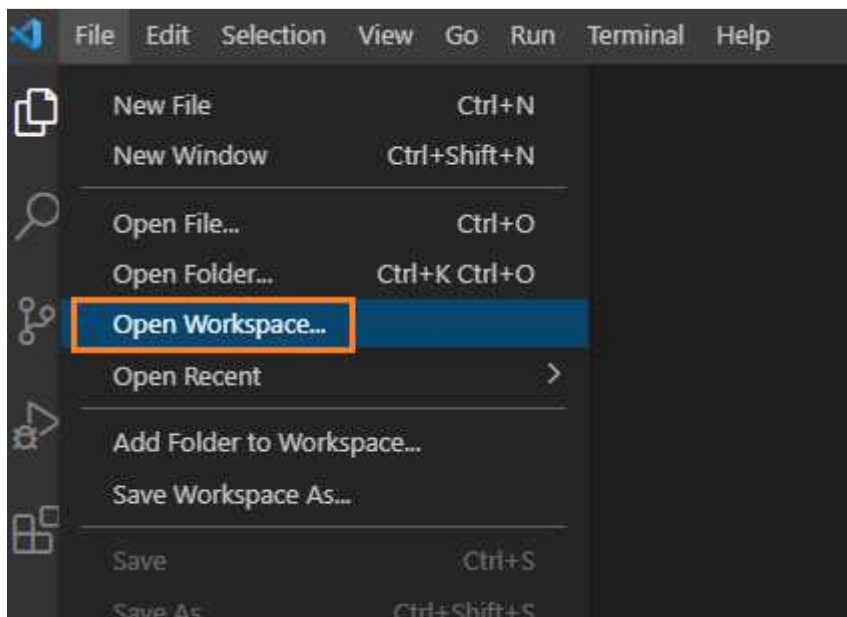
2. Objectifs

- ✓ Utiliser le module **FormsModule**.
- ✓ Créer un formulaire.
- ✓ Utiliser **l'évent binding** pour implémenter le clic sur un bouton.
- ✓ Utiliser la directive **ngModel** sur un composant HTML pour accéder à l'objet JavaScript qui le représente.
- ✓ Utiliser la directive **ngSubmit** pour associer une méthode TS au submit d'un formulaire.
- ✓ Utiliser la directive **ngForm** sur un formulaire pour accéder à l'objet JavaScript qui le représente.
- ✓ Utiliser la directive **ngIf**.
- ✓ Valider les données saisies au niveau du formulaire en affichant les erreurs le cas échéant.

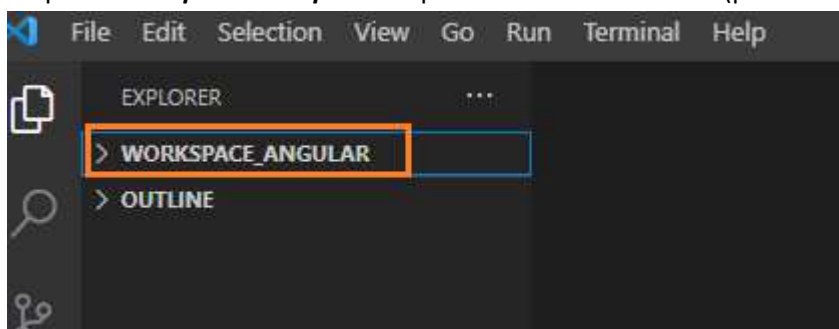
3. Développement

a. Création de l'application tpformualires

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



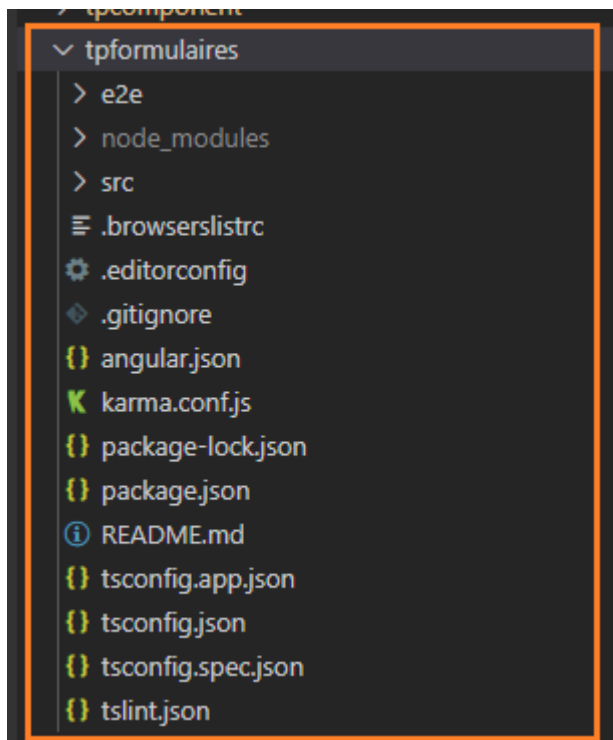
- Cliquer sur « **Open Workspace...** » puis choisir votre dossier (par exemple c:/workspace_angular) :



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

```
PS C:\workspace_Angular\tpdi> ng n tpformulaires
? 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 tpformulaires/angular.json (3679 bytes)
CREATE tpformulaires/package.json (1203 bytes)
```

- L'application angular **tpformulaires** ensuite générée par Angular CLI.



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

b. Création du composant Inscription

- Dans le terminal, lancer les commandes suivantes :

```
PS C:\workspace_Angular> cd .\tpformulaires\
PS C:\workspace_Angular\tpformulaires>
PS C:\workspace_Angular\tpformulaires> ng g c inscription
CREATE src/app/inscription/inscription.component.html (26 bytes)
CREATE src/app/inscription/inscription.component.spec.ts (661 bytes)
CREATE src/app/inscription/inscription.component.ts (295 bytes)
CREATE src/app/inscription/inscription.component.css (0 bytes)
UPDATE src/app/app.module.ts (495 bytes)
```

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

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { InscriptionComponent } from './inscription/inscription.component';

@NgModule({
  declarations: [
    AppComponent,
    InscriptionComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

d. Modification de la page inscription.component.html

```
<h1>Inscription</h1>
<div class="container">
  <form>

    <div class="form-group">
      <label for="firstname">Firstname</label><br>
      <input type="text" name="firstname" class="form-control">
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="lastname">Lastname</label><br>
      <input type="text" name="lastname" class="form-control">
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="email">Email</label><br>
      <input type="text" name="email" class="form-control">
  </form>
</div>
```

```

    </div>
    <pre></pre>

    <div class="form-group">
      <label for="password">Password</label><br>
      <input type="text" name="password" class="form-control">
    </div>
    <pre></pre>

    <button class="btn btn-primary" type="submit">submit</button>
  </form>
</div>

```

e. Modification de la feuille de style inscription.component.html

```

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

```

f. Modification de la page app.component.html

```
<inscription></inscription>
```

- Lancer L'URL : <http://localhost:4200> :

Inscription

Firstname

Lastname

Email

Password

g. La directive ngModel

- Ouvrir la page *inscription.component.html* et ajouter la directive **ngModel** au niveau des `<input>` :

```
<h1>Inscription</h1>
<div class="container">
  <form>

    <div class="form-group">
      <label for="firstname">Firstname</label><br>
      <input type="text" name="firstname" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="lastname">Lastname</label><br>
      <input type="text" name="lastname" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="email">Email</label><br>
      <input type="text" name="email" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="password">Password</label><br>
      <input type="password" name="password" class="form-control" ngModel>
    </div>
  </form>
</div>
```

```

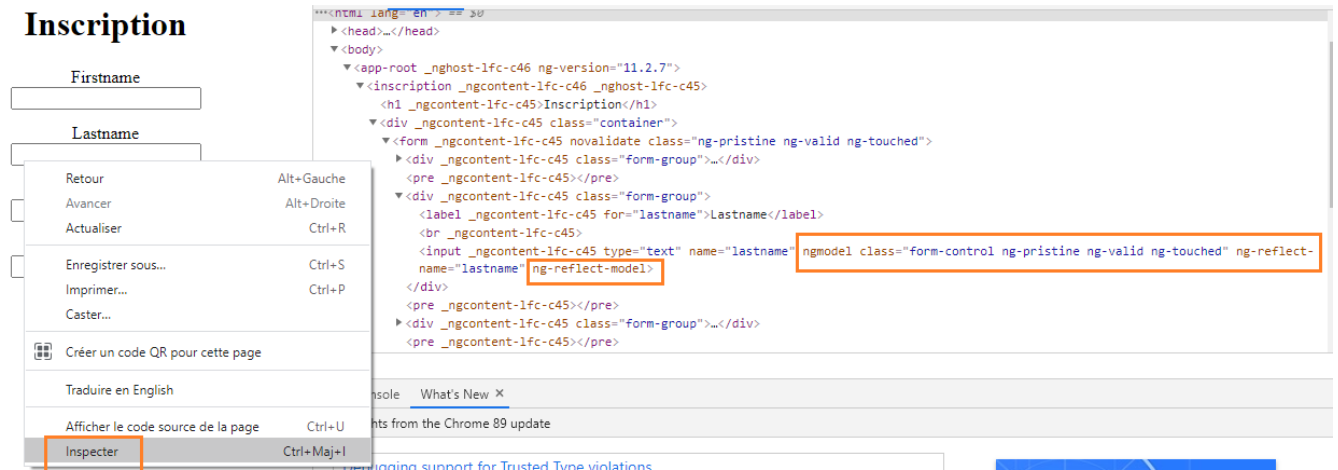
</pre></pre>

<button class="btn btn-primary" type="submit">submit</button>
</form>

</div>

```

- Au niveau du formulaire qui est affichée sur le navigateur, cliquer à droite de la souris sur Inspecter :



Vous constatez qu'au niveau du code HTML, Angular a reconnu la directive **ngModel** et a ajouté au niveau du formulaire les « Form control » suivants : **form-control ng-pristine ng-valid ng-touched**.

- Ajouter la directive **ngSubmit** au niveau de la balise form comme suit :

```

<h1>Inscription</h1>
<div class="container">
  <form (ngSubmit)="submit()">

    <div class="form-group">
      <label for="firstname">Firstname</label><br>
      <input type="text" name="firstname" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="lastname">Lastname</label><br>
      <input type="text" name="lastname" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="email">Email</label><br>

```



```

        <input type="text" name="email" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
        <label for="password">Password</label><br>
        <input type="text" name="password" class="form-control" ngModel>
    </div>
    <pre></pre>

    <button class="btn btn-primary" type="submit">submit</button>
</form>
</div>

```

- Ajouter la méthode submit() au niveau de la classe **InscriptionComponent** :

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'inscription',
  templateUrl: './inscription.component.html',
  styleUrls: ['./inscription.component.css']
})
export class InscriptionComponent {

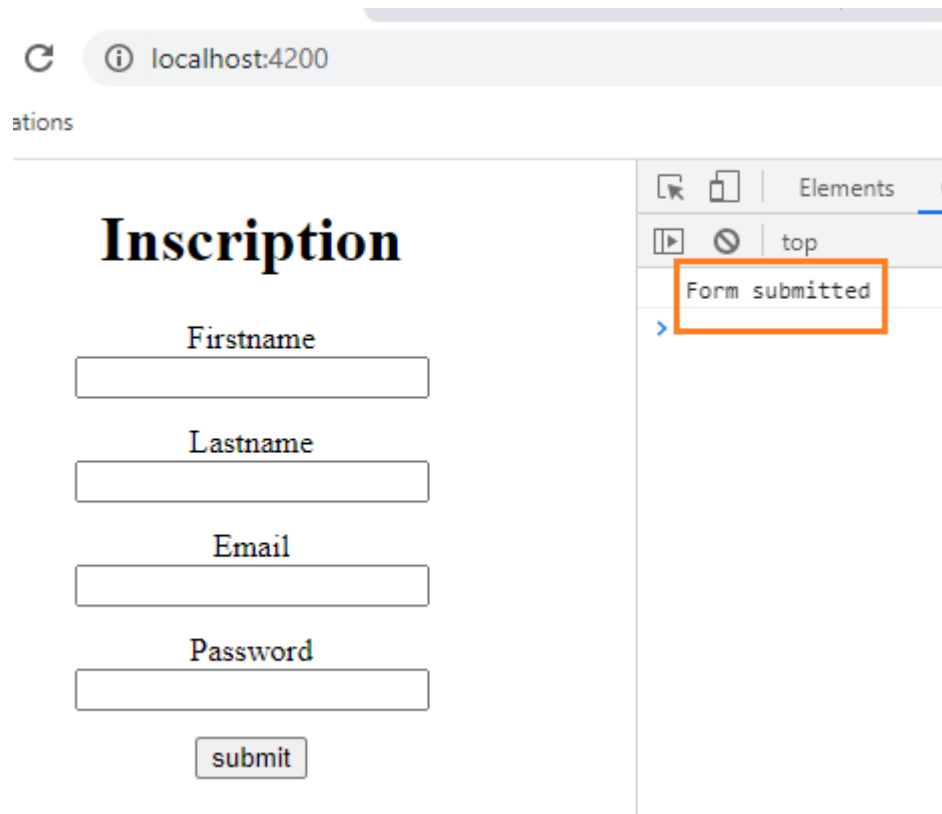
  constructor() { }

  submit(){
    console.log("Form submitted");
  }

}

```

- Cliquer ensuite sur le bouton submit et vérifier que la méthode console a été bien exécutée :



h. Présentation JavaScript du formulaire

- Ajouter la directive **ngForm** au niveau de la balise `<form>` et passer le nom de la directive à la méthode `submit()` comme suit :

```
<h1>Inscription</h1>
<div class="container">
  <form #login="ngForm" (ngSubmit)="submit(login)">

    <div class="form-group">
      <label for="firstname">Firstname</label><br>
      <input type="text" name="firstname" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="lastname">Lastname</label><br>
      <input type="text" name="lastname" class="form-control" ngModel>
```

```

    </div>
    <pre></pre>

    <div class="form-group">
      <label for="email">Email</label><br>
      <input type="text" name="email" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="password">Password</label><br>
      <input type="text" name="password" class="form-control" ngModel>
    </div>
    <pre></pre>

    <button class="btn btn-primary" type="submit">submit</button>
  </form>
</div>

```

- Ensuite, modifier le code de la méthode submit() comme suit :

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'inscription',
  templateUrl: './inscription.component.html',
  styleUrls: ['./inscription.component.css']
})
export class InscriptionComponent {

  constructor() { }

  submit(login:any){
    console.log("Form submitted",login);
  }

}

```

- Vérifier que la représentation JavaScript du formulaire :

Inscription

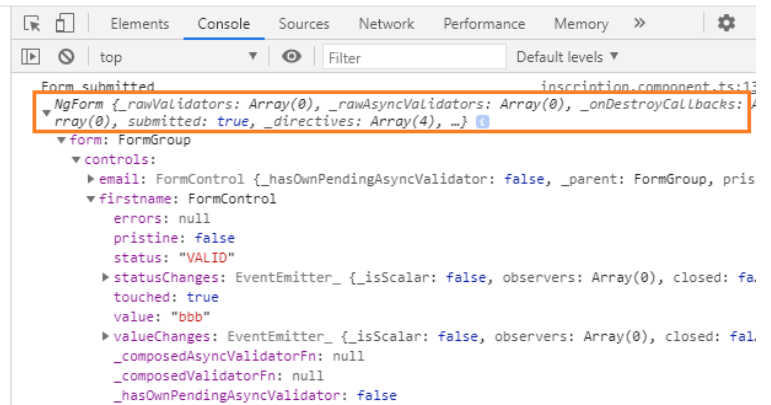
Firstname
bbb

Lastname

Email

Password

submit



i. Ajouter des règles de validation

- Modifier la page `inscription.component.html` comme suit :

```
<h1>Inscription</h1>
<div class="container">
  <form #login="ngForm" (ngSubmit)="submit(login)">

    <div class="form-group">
      <label for="firstname">Firstname</label><br>
      <input #firstname="ngModel" required minlength="2" maxlength="7" 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"> Firstname ne peut pas être vide</p>
        <p *ngIf="firstname.errors != null && firstname.errors.minlength"> firstname doit avoir au moins 2 caractères</p>
      </div>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="lastname">Lastname</label><br>
      <input type="text" name="lastname" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
      <label for="email">Email</label><br>
      <input type="text" name="email" class="form-control" ngModel>
    </div>
    <pre></pre>

    <div class="form-group">
```

```
        <label for="password">Password</label><br>
        <input type="text" name="password" class="form-control" ngModel>
    </div>
    <pre></pre>

    <button class="btn btn-primary" type="submit">submit</button>
</form>

</div>
```

- Tester votre formulaire :
 - Si vous touchez au champ firstname et vous ne saisissez rien, le message suivant est affiché :

Inscription

Firstname

Firstname ne peut pas être vide

Lastname

Email

Password

submit

- Si vous touchez au champ firstname et vous saisissez une valeur dont le nombre de caractères est inférieure à 2, le message suivant est affiché :

Inscription

Firstname

firstname doit avoir au moins 2 caractères

Lastname

Email

Password

Fin du TP 4.