

Formation Angular 11

TP N°8: Reactive Form avec Angular

SOMMAIRE

1.	Pré-requis :	3
2.	Objectifs	3
3.	Développement	3
a.	Création de l'application.....	3
b.	Créer le composant inscription.....	3
c.	Importer FormsModule et ReactiveFormsModule au niveau du module principal.....	3
d.	Modifier la classe <i>inscription.component.ts</i>	4
e.	Modifier le Template <i>inscription.component.html</i>	5
f.	Modification du fichier <i>app.component.html</i>	6
g.	Créer une règle de validation	6
h.	Tester l'application	9

1. Pré-requis :

- Réaliser les TPs 1-7 en premier.

2. Objectifs

- ✓ Utiliser le module **ReactiveFormsModule** et **FormsModule**.
- ✓ Utiliser **FormGroup**, **FormControl** et **Validators** du module **@angular/forms**.
- ✓ Comment attaché une règle de contrôle aux champs d'un formulaire.
- ✓ Comment utiliser la directive **formGroup**.
- ✓ Comment accéder aux contrôles gérés par le modèle au niveau du Template.
- ✓ Comment personnaliser votre propre règle de contrôle avec **AbstractControl** et **ValidationErrors**.

3. Développement

a. Création de l'application

- Créer l'application **tpreactiveform** :

```
PS C:\workspace_Angular> ng n tpreactiveform
? 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 tpreactiveform/angular.json (3687 bytes)
```

- Installer bootstrap et jquery au niveau de votre application (voir le TP 6).
- Démarrer ensuite l'application : **ng serve**.

b. Créer le composant inscription

- Lancer la commande : **ng g c inscription**

c. Importer **FormsModule** et **ReactiveFormsModule** au niveau du module principal

- Ajouter **FormsModule** et **ReactiveFormsModule** au niveau du module **app.module.ts** :

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

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

```

d. Modifier la classe *inscription.component.ts*

```

import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { CustomValidator } from '../validator/custom-validator';

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

  constructor() { }

  ngOnInit(): void {
  }

  form = new FormGroup({
    name: new FormControl('', [Validators.required, Validators.minLength(3)]),
    email: new FormControl('', [Validators.required, Validators.email]),
    phone: new FormControl('', [Validators.required]),
    comment: new FormControl('', Validators.required)
  });

  get f() {
    return this.form.controls;
  }

  submit() {
    if (this.form.status === 'VALID') {
      console.log(this.form.value);
    }
  }
}

```

```

setValue() {
  this.form.setValue({ name: 'ABBOU HASSANE', email: 'abbouhassane@gmail.com', phone:
'', comment: 'Exemple pour Reactive Forms' });
}

resetValue() {
  this.form.reset({ name: '', email: '', phone: '', comment: '' });
}
}

```

e. Modifier le Template inscription.component.html

```

<h1>Formulaire d'inscription :</h1>

<form [formGroup]="form" (ngSubmit)="submit()">
  <div class="col-sm-4 col-sm-offset-4">
    <div class="form-group">
      <label for="name">Nom</label>
      <input formControlName="name" id="name" type="text" class="form-control">
      <div *ngIf="f.name.touched && f.name.invalid" class="alert alert-danger">
        <div *ngIf="f.name.errors != null && f.name.errors.required">Nom obligat
oire.</div>
        <div *ngIf="f.name.errors !=null && f.name.errors.minlength">Le nom doit
être composé au minimum de 3
        caractères.</div>
      </div>
    </div>

    <div class="form-group">
      <label for="email">Email</label>
      <input formControlName="email" id="email" type="text" class="form-control">
      <div *ngIf="f.email.touched && f.email.invalid" class="alert alert-danger">
        <div *ngIf="f.email.errors != null && f.email.errors.required">Email obl
igatoire.</div>
        <div *ngIf="f.email.errors != null && f.email.errors.email">Veuillez sai
sir une adresse mail correct.
      </div>
    </div>

    <div class="form-group">
      <label for="phone">Phone</label>
      <input formControlName="phone" id="phone" type="text" class="form-control">
    </div>
  </div>

```

```

    <div class="form-group">
      <label for="comment">Commentaire</label>
      <textarea formControlName="comment" id="comment" type="text" class="form-
control">
    </textarea>
      <div *ngIf="f.comment.touched && f.comment.invalid" class="alert alert-
danger">
        <div *ngIf="f.comment.errors != null && f.comment.errors.required">Le co
mmentaire est obligatoire.</div>
      </div>
    </div>
  </div>
  <button class="btn btn-
info" type="button" (click)="setValue()">Initialiser les valeurs par défaut</button>
  <button class="btn btn-info" type="button" (click)="resetValue()">Effacer</button>
  <button class="btn btn-primary" type="submit">Submit</button>
</form>

```

f. Modification du fichier app.component.html

```
<app-inscription></app-inscription>
```

g. Créer une règle de validation

- Créer la classe CustomValidator : **ng g class validator/CustomValidator**
- Modifier le fichier **custom-validator.ts** comme suit :

```

import { AbstractControl, ValidationErrors } from "@angular/forms";

export class CustomValidator {
  static cannotContainSpace(control: AbstractControl) : ValidationErrors | null {
    if((control.value as string).indexOf(' ') >= 0){
      return {cannotContainSpace: true}
    }
    return null;
  }

  static ValidatePhone(control: AbstractControl): {[key: string]: any} | null {
    if (control.value && control.value.length != 10) {
      return { phoneNumberInvalid: true };
    }
    return null;
  }
}

```

```
}  
}
```

Ajouter ces deux règles de contrôle au niveau de contrôles de vos données (classe *InscriptionComponent*):

```
import { Component, OnInit } from '@angular/core';  
import { FormControl, FormGroup, Validators } from '@angular/forms';  
import { CustomValidator } from '../validator/custom-validator';  
  
@Component({  
  selector: 'app-inscription',  
  templateUrl: './inscription.component.html',  
  styleUrls: ['./inscription.component.css']  
})  
export class InscriptionComponent implements OnInit {  
  
  constructor() { }  
  
  ngOnInit(): void {  
  
    form = new FormGroup({  
      name: new FormControl('', [Validators.required, Validators.minLength(3), CustomValidator.cannotContainSpace]),  
      email: new FormControl('', [Validators.required, Validators.email]),  
      phone: new FormControl('', [Validators.required, CustomValidator.ValidatePhone]),  
      comment: new FormControl('', Validators.required)  
    });  
  
    get f() {  
      return this.form.controls;  
    }  
  
    submit() {  
      if (this.form.status === 'VALID') {  
        console.log(this.form.value);  
      }  
    }  
  
    setValue() {  
      this.form.setValue({ name: 'ABBOU HASSANE', email: 'abbouhassane@gmail.com', phone: '', comment: 'Exemple pour Reactive Forms' });  
    }  
  
    resetValue() {
```

```

    this.form.reset({ name: '', email: '', phone: '', comment: '' });
  }
}

```

- Modifier le Template inscription.component.html comme suit :

```

<h1>Formulaire d'inscription :</h1>

<form [formGroup]="form" (ngSubmit)="submit()">
  <div class="col-sm-4 col-sm-offset-4">
    <div class="form-group">
      <label for="name">Nom</label>
      <input formControlName="name" id="name" type="text" class="form-
control">
      <div *ngIf="f.name.touched && f.name.invalid" class="alert alert-
danger">
        <div *ngIf="f.name.errors != null && f.name.errors.required">Nom o
bligatoire.</div>
        <div *ngIf="f.name.errors !=null && f.name.errors.minlength">Le no
m doit être composé au minimum de 3
        caractères.</div>
        <div *ngIf="f.name.errors !=null && f.name.errors.cannotContai
nSpace">Le nom ne doit pas contenir des espaces.</div>
      </div>
    </div>

    <div class="form-group">
      <label for="email">Email</label>
      <input formControlName="email" id="email" type="text" class="form-
control">
      <div *ngIf="f.email.touched && f.email.invalid" class="alert alert-
danger">
        <div *ngIf="f.email.errors != null && f.email.errors.required">Ema
il obligatoire.</div>
        <div *ngIf="f.email.errors != null && f.email.errors.email">Veuell
ez saisir une adresse mail correct.
        </div>
      </div>
    </div>

    <div class="form-group">
      <label for="phone">Phone</label>
      <input formControlName="phone" id="phone" type="text" class="form-
control">
      <div *ngIf="f.phone.touched && f.phone.invalid" class="alert alert-
danger">
        <div *ngIf="f.phone.errors != null && f.phone.errors.phoneNumberIn

```



```

valid">Le numéro de téléphone doit être composé de 10 chiffres.
    </div>
  </div>
</div>

  <div class="form-group">
    <label for="comment">Commentaire</label>
    <textarea formControlName="comment" id="comment" type="text" class="form-control">
    </textarea>
    <div *ngIf="f.comment.touched && f.comment.invalid" class="alert alert-danger">
      <div *ngIf="f.comment.errors != null && f.comment.errors.required">
        >Le commentaire est obligatoire.</div>
      </div>
    </div>
  </div>
  <button class="btn btn-info" type="button" (click)="setValue()">Initialiser les valeurs par défaut</button>
  <button class="btn btn-info" type="button" (click)="resetValue()">Effacer</button>
  <button class="btn btn-primary" type="submit">Submit</button>
</form>

```

h. Tester l'application

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

The screenshot shows a web browser window with the address bar set to 'localhost:4200'. The page title is 'Applications'. The main heading is 'Formulaire d'inscription :'. Below the heading are four input fields labeled 'Nom', 'Email', 'Phone', and 'Commentaire'. At the bottom of the form are three buttons: 'Initialiser les valeurs par défaut' (light blue), 'Effacer' (light blue), and 'Submit' (dark blue).

- Veuillez entrer les données et vérifier les messages d'erreur :

Formulaire d'inscription :

Nom

jj jjjj

Le nom ne doit pas contenir des espaces.

Email

jj

Veuillez saisir une adresse mail correct.

Phone

222

Le numéro de téléphone doit être composé de 10 chiffres.

Commentaire

jjj

Initialiser les valeurs par défaut

Effacer

Submit

Vous venez de développer un formulaire Reactive !

Vous pouvez passer au TP n° 9.

Fin du TP 9.