

Approche des Formulaires avec Angular

Template-Driven Forms

Description :

Basées sur des directives **Angular** et sont principalement gérées dans le template **HTML**. Elles sont **simples** à mettre en place et conviennent bien pour des **formulaires simples**.

Avantages

- Facile à mettre en place pour des formulaires simples.
- Moins de code **TypeScript** à écrire.
- Utilisation de directives comme **ngModel** pour la liaison de données bidirectionnelle.

Inconvénients

- Moins flexible pour des formulaires complexes.
- Difficile à tester unitairement.
- Moins de contrôle sur la logique de validation et de gestion des erreurs.

Exemple :

Code la vue

```
<form #myForm="ngForm" (ngSubmit)="onSubmit(myForm)">
  <input name="username" ngModel required>
  <button type="submit">Submit</button>
</form>
```

Code de la logique

```
export class MyFormComponent {
  // Variable pour stocker la valeur du champ "username"
  submittedUsername: string | null = null;

  // Méthode appelée lors de la soumission du formulaire
  onSubmit(form: NgForm) {
    // Vérifie si le formulaire est valide
    if (form.valid) {
      // Récupère la valeur du champ "username"
      this.submittedUsername = form.value.username;
      console.log('Nom :', this.submittedUsername);
    }
  }
}
```

Reactive Forms

Description :

Les Reactive Forms sont basées sur une approche programmatique, où le formulaire est défini et géré dans la logique du composant. Ils offrent plus de contrôle et de flexibilité.

Avantages

- Plus de contrôle sur la logique du formulaire.
- Facile à tester unitairement.
- Gestion fine des validations et des erreurs.
- Possibilité de créer des formulaires dynamiques.

Inconvénients

- Plus de code à écrire par rapport aux Template-Driven Forms.
- Courbe d'apprentissage légèrement plus élevée.

Exemple :

Code la vue

```
<form [formGroup]="myForm" (ngSubmit)="onSubmit()">
  <input formControlName="username">
  <button type="submit">Submit</button>
</form>
```

Code de la logique

```
export class MyComponent {
  myForm: FormGroup;
  constructor(private fb: FormBuilder) {
    this.myForm = this.fb.group({
      username: ['', Validators.required]
    });
  }
  onSubmit() {
    console.log(this.myForm.value);
  }
}
```