



1

Créer un formulaire template dans le HTML

2

Utiliser ngModel pour lier les champs aux variables

3

Construire un FormGroup avec FormBuilder

4

Associer des Validators aux champs de formulaire

5

Observer valueChanges pour réagir aux entrées

Code

```

1 // Template form avec ngModel
2 <input type="email" name="userEmail" [(ngModel)]="userEmail" required>
3
4 // Récupération d'un NgForm complet
5 <form #newsletterForm="ngForm" (ngSubmit)="onSubmitForm(newsletterForm)">
6
7 // Reactive form : déclaration d'un FormGroup
8 contactForm = this.formBuilder.group({
9   name: [''],
10  email: [''],
11  course: [''],
12  message: ['']
13 });
14
15 // Validation d'un champ email
16 email: [''], [Validators.required, Validators.email]]
17
18 // Observation d'un champ avec valueChanges
19 selectedCourse$ = this.contactForm.valueChanges.pipe(
20   map(form => form.course),
21   filter(course => course !== ''),
22   distinctUntilChanged()
23 );

```

Définitions

Formulaire template

Formulaire défini dans le HTML et piloté par Angular via ngModel et ngForm.

Formulaire réactif

Formulaire construit en TypeScript avec FormBuilder, offrant un contrôle avancé.

Two-way binding

Liaison bidirectionnelle entre le modèle et la vue, réalisée avec [(ngModel)].

NgForm

Objet représentant un formulaire template Angular, accessible via une référence locale.

Validators

Fonctions Angular permettant de définir des règles de validation sur les champs d'un formulaire.

Bonnes pratiques

- Utiliser ngSubmit pour réagir à la soumission
- Ajouter FormsModule ou ReactiveFormsModule selon le besoin
- Appliquer des Validators dès la configuration des champs
- Utiliser ngModel pour le two-way binding
- Afficher un message si le formulaire est invalide
- Structurer les champs avec FormControlName cohérents
- Souscrire proprement aux Observables
- Utiliser updateOn: 'blur' pour limiter les émissions

Erreurs classiques

- Oublier d'ajouter name sur les inputs d'un formulaire template
- Appeler les Validators au lieu de les passer comme fonction
- Souscrire sans gérer la désinscription
- Utiliser des noms différents entre le FormControlName et le FormGroup
- Valider uniquement côté front sans revalider côté serveur
- Omettre l'import de ReactiveFormsModule ou FormsModule
- Afficher des aperçus d'URL avant leur complétion
- Négliger le feedback utilisateur en cas d'erreur