

## Workshop n°8 :

### Template Driven Forms

#### Les objectifs :

- Créer un formulaire avec l'approche Template Driven Form.
- Ajouter des validateurs
- Gérer les messages d'erreurs.

#### Le rendu final :

Créer le formulaire suivant permettant la participation à un évènement :

Exemple : <http://localhost:4200/events/participate/1/50>

Adresse e-mail

Nombre de places

Prix total : 200

**Soumettre**

**Figure 1 : participationFormComponent : Formulaire de participation**

#### Les instructions à suivre :

##### A- *Création de formulaire*

- 1- Dans le composant ListEventCompenent, ajouter un bouton « Participer » comme indiqué dans la figure 2.



**Figure 2 : ListEventComponent : bouton « Proposer un évènement »**

- 2- Générer le composant participationFormComponent sous features/events/components : Taper la commande : **ng g c features/events/components /participationForm -skip-tests**
- 3- Dans EventsRoutingModule, ajouter une route menant vers le composant **participationFormComponent** en envoyant dans l'url l'id de l'évènement et son prix.
- 4- En cliquant sur le bouton « Participer » une redirection se fait vers le composant **participationFormComponent**. Utiliser le service « **Router** » pour faire la redirection dans une méthode participate() à créer dans le composant **eventCardComponent**.
- 5- Créer le formulaire permettant la participation à un évènement comme indiqué plus haut dans la figure 1 selon l'approche Template Driven Forms.
- 6- Créer un nouveau modèle de donnée appelé « Participation » comme suit :

```
export class Participation {
  id!: number;
  userId: number;
  eventId: number;
  emailParticipant: string;
  nbPlaces:number;
  status: 'confirmed' | 'pending' | 'cancelled';
  registrationDate: Date;
  constructor(
    userId: number,
    eventId: number,
    emailParticipant:string,
    nbPlaces:number,
    status: 'confirmed' | 'pending' | 'cancelled',
  ) {
    this.userId = userId;
    this.eventId = eventId;
  }
}
```

```

this.emailParticipant=emailParticipant;
this.nbPlaces=nbPlaces;
this.status = status;
this.registrationDate = new Date();
}

```

## **B- Validation de formulaire – Validateurs prédefinies**

- 7- En tenant compte des remarques cité plus haut, ajouter les validateurs suivants au niveau du formulaire :
  - Tous les champs sont obligatoires.
  - Le champ « email » doit respecter le format email.
  - Le champ « Nombre de place » est un nombre supérieur strictement à 0
  - Le bouton « Soumettre » est désactivé tant que le formulaire est INVALID .
- 8- Ajoutez un message d’erreur bien précis associé à chaque validateur.

Adresse e-mail

L'adresse e-mail est obligatoire.

Nombre de places

Le nombre de places doit être un nombre valide et supérieur à 0.

**Soumettre**

**Figure 3 : Messages d’erreurs**

- 9- Quand le nombre de place est valide et l’utilisateur quitte le champ de saisi, un message indiquant le prix total est affiché en bas comme indiqué dans la figure 1.  
NB : Le prix total est affiché selon le nombre de places saisi dans le formulaire et selon le prix de l’évènement envoyé dans l'url.
- 10- Une fois le bouton « Soumettre » est actif, en cliquant dessus, les informations sont saisies au niveau du formulaire sont ajoutées dans la liste des participations à définir dans le composant.