

## Workshop n°9-suite :

### Manipulation des méthodes http

### POST – PUT - DELETE

#### Les objectifs :

- Créer et injecter un service.
- Consommer des API REST à travers le service HttpClient
- Manipuler différentes méthodes du service HttpClient : GET, POST, PUT, DELETE

**Le rendu final :** Une application dynamique.

#### A- Envoi des données vers le backend « Méthode POST »

- 1- Au niveau du service EventsService créer la méthode « addEventToBackend() » qui permet d'ajouter un évènement dans votre backend (dans notre cas, dans le fichier db.json)
- 2- Au niveau du composant addEventComponent crée dans le workshop n°7 « Réactive Form », appeler cette nouvelle méthode dans le bon endroit.

NB : Un évènement bien ajouté doit être de **la forme suivante** si vous avez finalisé tout le workshop de « Reactive Form »

```
{
  "id": 3
  "title": "MyEvent",
  "description": "Description de ma MyEvent.  ",
  "date": "2025-12-07",
  "price": 25,
  "nbPlaces": 50,
  "organizerId": 1,
  "imageUrl": "image1.png",
  "nbLikes": 0,
  "address": {
    "street": "Mohame Ali",
    "city": "Ariana soghra",
    "state": "Ariana",
    "zip": "7052"
  },
  "domaines": [
    "web",
    "fintech"
  ]
}
```

- 3- Ajouter une redirection vers un nouveau composant à créer appelé **MyEventsComponent** qui affiche la liste des évènements que tu as ajouté.

Nous supposons que ton id qui correspond à l'attribut **organizerId= 1**.

**NB : Pensez à créer une méthode du service qui retourne la liste des évènements pour un organisateur bien déterminé.**

**Exemple : localhost:3000/events?organizerId=1**

« Remarque : Vous pouvez travailler avec le composant ListEventsComponent mais il faut gérer le cas où ce composant affiche tout les évènements et le cas où ce composant affiche uniquement les évènements d'un organisateur bien déterminé. »

## B- Modification des données « Méthode PUT »

- 4- Dans le composant MyEventsComponent, ajouter un bouton « modifier » qui permet de mettre à jour un événement. Le click sur le bouton permet d'afficher le formulaire du composant addEventComponent contenant initialement les informations d'un événement.
- 5- Une fois vous cliquez sur le bouton « Soumettre », les données seront modifiées dans la base de données (fichier db.json).

## C- Suppression des données « Méthode DELETE »

- 6- Dans le composant MyEventsComponent, ajouter un bouton « supprimer » qui permet de supprimer un événement.
- 7- Une fois vous cliquez sur le bouton « Supprimer », l'évènement est supprimé de la base de données (fichier db.json).

## Travail à faire :

### Ajout d'une participation

- 8- Au niveau du service EventsService créer la méthode « addParticipationToBackend() » qui permet d'ajouter une participation dans votre backend (dans notre cas, dans le fichier db.json)
- 9- Ajouter une redirection vers un nouveau composant à créer appelé MyParticipationsComponent qui affiche la liste des participations que tu as ajoutés.  
**Nous supposons que ton userId = 1.**

### Suppression d'une participation

- 10- Dans le composant « MyParticipationsComponent », ajouter un bouton « annuler ».
- 11- En cliquant sur ce bouton la participation correspondante est supprimée.

### Gestion des erreurs

- 12- Gérer les erreurs qui pourront être commises suite à l'appel d'une API à votre choix (faire une redirection, afficher une alerte, MatSnackBar.....)