

Test Pratique Angular : Gestion de Réservation de Trottinettes

Durée : 45 minutes

Technologies : Angular, JSON-Server

Contexte

Vous devez développer une petite application Angular permettant la gestion des réservations de trottinettes. Chaque trottinette est identifiée par : **idTrottinette**, **modèle**.

Chaque réservation contient : **id réservation**, **email**, **téléphone**, **date de début**, **nombre d'heures**, **idTrottinette**.

L'application doit permettre de :

1. Lister les trottinettes disponibles.
 2. Sélectionner une trottinette et ajouter une réservation en saisissant les informations demandées.
-

Configuration initiale

1. Installez et configurez json-server pour simuler une API REST. Utilisez un fichier db.json contenant la liste des trottinettes et des réservations.

Exemple de données :

```
{
  "trottinettes": [
    { "id": 1, "modele": "Xiaomi Pro 2" },
    { "id": 2, "modele": "Ninebot Max G30" }
  ],
  "reservations": []
}
```

Travail demandé

1. **Créer les composants suivants :**
 - HeaderComponent : contient les liens suivants : Home, Trottinettes, Réservation
 - TrotinetListComponent : Afficher la liste des trottinettes disponibles et inclure un bouton "Réserver" pour chaque trottinette.

- ReservationFormComponent : Formulaire permettant de réserver une trottinette.
- ReservationComponent : Afficher la liste des trottinettes.

2. Configurer le fichier **app-routing.module.ts** avec les routes nécessaires qui permet :

- Afficher la liste des trottinettes sur la page d'accueil dès le lancement de l'application
- Afficher la liste des trottinettes (trottinettes).
- Afficher la liste des réservations (reservations).
- Permettre l'ajout d'une nouvelle réservation via un formulaire (add reservation).
- Passer l'ID de la trottinette sélectionnée en paramètre au formulaire de réservation.

3. Créer le formulaire de réservation :

- Champs requis :
 - ID de la réservation (obligatoire).
 - ID trottinette prérempli avec la valeur passée en paramètre.
 - Email (obligatoire).
 - Téléphone (obligatoire),
 - La date de réservation est obligatoire et doit respecter le pattern suivant `^\d{4}-\d{2}-\d{2}$`
 - Nombre d'heures (obligatoire et minimum **1 heure**).
- Valider les champs avec les contraintes suivantes :
 - Tous les champs doivent être remplis.
 - La date doit respecter le format requis
 - La durée de réservation doit être d'au moins une heure.
- Bouton "Valider" en bas du formulaire initialement désactivé.

4. Créer les services suivants :

- **TrottinetteServices**
getTrottinettes() : Récupérer la liste des trottinettes.

- **RéservationServices**

getReservations() : Récupérer la liste des réservations.

addReservation(reservation) : Ajouter une nouvelle réservation.

Interfaces proposées

1. Liste des trottinettes

- Affichage des trottinettes sous forme de cartes.
- Bouton "Réserver" à côté de chaque trottinette.
- Affichage des réservations sous forme d'un tableau.