

# Exercice 3

## Description de l'Exercice

Vous allez ajouter une nouvelle entité `Ecole` qui aura une relation One-to-Many avec l'entité `Etudiant`. Les étudiants doivent être mis à jour pour inclure une référence à une école, mais cette relation doit rester unidirectionnelle dans les DTO, c'est-à-dire qu'un étudiant ne pourra pas accéder à son école via les DTO.

## Étapes à Suivre

### 1. Créer l'entité `Ecole`

- **Champs :**
  - `id` (`Long`)
  - `nom` (`String`)
  - `domaine` (`String`)
  - `etudiants` (`List<Etudiant>`)
- Annoter l'entité `Ecole` avec les annotations JPA nécessaires pour établir une relation One-to-Many avec `Etudiant`.

### 2. Mettre à jour l'entité `Etudiant`

- Ajouter un champ `ecole` de type `Ecole` dans l'entité `Etudiant`.
- Annoter ce champ avec `@ManyToOne` et `@JoinColumn`.

### 3. Créer les DTOs

- Créer le DTO pour.
  - `EcoleDTO` : Inclure tous les champs de l'entité `Ecole` sauf la liste des étudiants.

### 4. Configurer MapStruct

- Configurer MapStruct pour mapper entre les entités et les DTOs.

## 5. Créer les Repositories

- Créer le repository pour l'entité `Ecole`.

## 6. Services

Implémentez les méthodes suivantes dans la classe `EcoleService` :

- `findAll()`: retourne toutes les écoles avec leurs étudiants
- `findById(Long id)`: retourne une écoles avec ses etudiants
- `save(EcoleDto ecoleDto)`: sauvegarde en base une écoles avec une liste d'étudiant. Retour aussi l'écoles sous forme de DTO.

Mettez aussi à jour `EtudiantService` pour prendre en compte l'association entre etudiant et ecole.

## 7. Controllers

Il faut ensuite créer un controller pour `Ecole`, avec les endpoints suivants :

- `GET /ecoles` : retourne la liste de toutes les écoles en appelant la méthode `findAll()` du service
- `GET /ecoles/{id}`: retourne les details d'une écoles spécifique par ID en appelant la méthode `findById(Long id)` du service.
- `POST/ecoles` : permet d'enregistrer une école avec ses étudiants, en utilisant `save(EcoleDto ecoleDto)` du service.

Instructions Supplémentaires :

Réaliser des tests via postman, et bien verifier avec swagger que tout est fonctionnel.