TP JPA 02 : Associations avec Jakarta Persistence

Gestion des relations entre entités JPA

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-11-01

Table des matières

1 Créer un nouveau projet Java JPA 1
2 Partie 1 - Entités de base 1
3 Partie 2 - Associations complexes 2
4 Partie 3 - Génération de données (1h) 2

1 Créer un nouveau projet Java JPA

Par exemple avec la structure suivante :

```
src/
├── main/java/fr/univtln/.../tpjpa02/
│   ├── entities/
│   ├── repositories/
│   └── utils/
└── test/java/fr/univitln/.../tpjpa02/
```

Le répertoire `entities` contiendra les entités JPA, `repositories` les classes de gestion des entités et `utils` les classes utilitaires. Vous pourrez utilise H2 comme au TP précédent pour simplifier la configuration.

2 Partie 1 - Entités de base

Créer les entités suivantes avec leurs associations et tester les opérations CRUD. Vous ajouterez les classes nécessaires.

```
@Entity @Getter @Setter
public class Artist {
```

```
    @Id @GeneratedValue
    private Long id;
    private String name;

    @OneToMany(mappedBy = "artist", cascade = CascadeType.ALL)
    private Set<Album> albums = new HashSet<>();
}
```

```
@Entity @Getter @Setter
public class Album {
    @Id @GeneratedValue
    private Long id;
    private String title;
    private LocalDate releaseDate;

    @ManyToOne
    private Artist artist;
}
```

3 Partie 2 - Associations complexes

Compléter votre programme avec d'autres entités et des associations plus complexes. Créer les entités suivantes avec leurs associations:

1. `Label`
   - OneToOne avec Address
   - OneToMany avec Album
2. `Playlist`
   - ManyToOne avec User
   - ManyToMany avec Track
3. `Concert`
   - ManyToOne avec Artist
   - OneToMany avec Ticket
   - ManyToMany avec Genre
4. `Review`
   - ManyToOne avec User
   - ManyToOne avec Album
   - ElementCollection pour les commentaires

4 Partie 3 - Génération de données (1h)

Utiliser Java Faker pour générer des données de test réalistes. https://github.com/DiUS/java-faker

- http://dius.github.io/java-faker/apidocs/com/github/javafaker/Artist.html
- http://dius.github.io/java-faker/apidocs/com/github/javafaker/Music.html
- ...

Vous pouvez aussi utiliser un LLM (Language Level Model) pour générer des fichiers CSV de données et écrire des script d'importation.