

# Collections de base

## Démonstration 10 du module 5

L'objectif de cette démonstration est

- Mettre en place une collection de base

## Contexte

- Continuer dans le projet précédent
- Nouveau sujet : Une personne peut pratiquer plusieurs sports. Cette liste de sports, peut être vue comme une simple liste de String
- Renommer le package des entités précédentes en com.
- Mettre l'annotation @Profil(«Demo») sur le bean dans la classe d'exécution

## Déroulement

- Créer un package : fr.eni.demo.collections
- Créer la classe Personne

```
package fr.eni.demo.collections;

import java.util.ArrayList;
import java.util.List;

import javax.persistence.CollectionTable;
import javax.persistence.ElementCollection;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.Table;

@Entity(name = "personne_cb")
@Table(name = "personne_cb")
public class Personne {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    private String nom;
    private String prenom;

    @ElementCollection(fetch = FetchType.EAGER)
    @CollectionTable(name = "Sports", joinColumns = @JoinColumn(name = "id_spo", referencedColumnName = "id"))
    private List<String> sports;
```

```

public Personne() {
    sports = new ArrayList<String>();
}

public Personne(String nom, String prenom) {
    this();
    this.nom = nom;
    this.prenom = prenom;
}

public void addSport(String sport) {
    sports.add(sport);
}

public long getId() {
    return id;
}

public void setId(long id) {
    this.id = id;
}

public String getNom() {
    return nom;
}

public void setNom(String nom) {
    this.nom = nom;
}

public String getPrenom() {
    return prenom;
}

public void setPrenom(String prenom) {
    this.prenom = prenom;
}

public List<String> getSports() {
    return sports;
}

public void setSports(List<String> sports) {
    this.sports = sports;
}

@Override
public String toString() {
    return "Personne [id=" + id + ", nom=" + nom + ", prenom=" + prenom + ", sports=" + sports
+ " ]";
}
}

```

- Elle contient une List<String> qui représente les sports
  - Nous précisons qu'elle sera remontée en même temps que la personne : fetch = FetchType.EAGER
  - Et nous définissons la table associée :

```

@CollectionTable(name = "Sports", joinColumns = @JoinColumn(name = "id_spo",
referencedColumnName = "id"))

```

- Créer un Repository

```
package fr.eni.demo.collections;

import org.springframework.data.jpa.repository.JpaRepository;

public interface PersonneCBRepository extends JpaRepository<Personne, Long> {

}
```

- Dans la classe d'exécution de l'application
  - Copier le code du nouveau bean :

```
@Bean
public CommandLineRunner demoCollectionsBases(PersonneCBRepository persDAO) {
    return (args) -> {

        fr.eni.demo.collections.Personne albert = new fr.eni.demo.collections.Personne("Dupontel", "Albert");
        fr.eni.demo.collections.Personne sophie = new fr.eni.demo.collections.Personne("Marceau", "Sophie");

        albert.addSport("Athlétisme");
        albert.addSport("Judo");

        sophie.addSport("Football");
        sophie.addSport("Judo");
        sophie.addSport("Karaté");

        persDAO.save(albert);
        persDAO.save(sophie);

        System.out.println("Liste des personnes : ");
        System.out.println("-----");
        for (fr.eni.demo.collections.Personne personne : persDAO.findAll()) {
            System.out.println(personne.toString());
        }

    };
}
```

- Exécution, Traces attendues :

```
...
Hibernate:
    create table personne_cb (
        id bigint not null auto_increment,
        nom varchar(255),
        prenom varchar(255),
        primary key (id)
    ) engine=InnoDB
Hibernate:
    create table sports (
        id_spo bigint not null,
        sports varchar(255)
    ) engine=InnoDB
Hibernate:
    alter table sports
        add constraint FKpyrh469a5ijfh3m7but2bxuoj
        foreign key (id_spo)
        references personne_cb (id)
...
Hibernate:
    insert
    into
        personne_cb
```

```

        (nom, prenom)
    values
        (?, ?)
Hibernate:
    insert
    into
        sports
        (id_spo, sports)
    values
        (?, ?)
Hibernate:
    insert
    into
        sports
        (id_spo, sports)
    values
        (?, ?)
Hibernate:
    insert
    into
        personne_cb
        (nom, prenom)
    values
        (?, ?)
Hibernate:
    insert
    into
        sports
        (id_spo, sports)
    values
        (?, ?)
Hibernate:
    insert
    into
        sports
        (id_spo, sports)
    values
        (?, ?)
Hibernate:
    insert
    into
        sports
        (id_spo, sports)
    values
        (?, ?)
Liste des personnes :
-----
Hibernate:
    select
        personne0_.id as id1_0_,
        personne0_.nom as nom2_0_,
        personne0_.prenom as prenom3_0_
    from
        personne_cb personne0_
Hibernate:
    select
        sports0_.id_spo as id_spo1_1_0_,
        sports0_.sports as sports2_1_0_
    from
        sports sports0_
    where
        sports0_.id_spo=?
Hibernate:
    select
        sports0_.id_spo as id_spo1_1_0_,
        sports0_.sports as sports2_1_0_

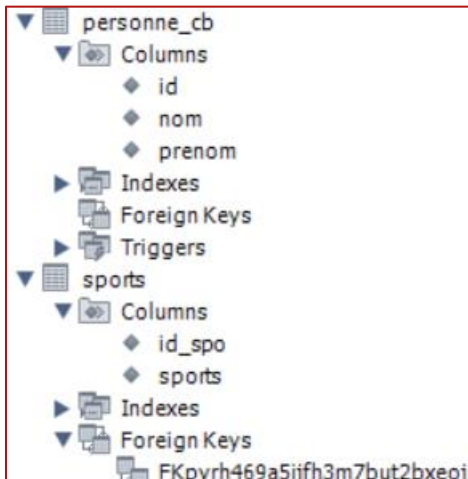
```

```

from
    sports sports0_
where
    sports0_.id_spo=?
Personne [id=1, nom=Dupontel, prenom=Albert, sports=[Athlétisme, Judo]]
Personne [id=2, nom=Marceau, prenom=Sophie, sports=[Football, Judo, Karaté]]

```

- Hibernate crée
  - Une table personne\_cb.
  - Une table sports
  - Une clef de jointure entre les 2.



- Voici les données en base :

Personne_cb	sports																					
<table><tr><th>id</th><th>nom</th><th>prenom</th></tr><tr><td>1</td><td>Dupontel</td><td>Albert</td></tr><tr><td>2</td><td>Marceau</td><td>Sophie</td></tr></table>	id	nom	prenom	1	Dupontel	Albert	2	Marceau	Sophie	<table><tr><th>id_spo</th><th>sports</th></tr><tr><td>1</td><td>Athlétisme</td></tr><tr><td>1</td><td>Judo</td></tr><tr><td>2</td><td>Football</td></tr><tr><td>2</td><td>Judo</td></tr><tr><td>2</td><td>Karaté</td></tr></table>	id_spo	sports	1	Athlétisme	1	Judo	2	Football	2	Judo	2	Karaté
id	nom	prenom																				
1	Dupontel	Albert																				
2	Marceau	Sophie																				
id_spo	sports																					
1	Athlétisme																					
1	Judo																					
2	Football																					
2	Judo																					
2	Karaté																					

- Possibilité d'enregistrer une collection d'éléments simple (String, Date, Integer...) sans avoir besoin de créer une nouvelle classe Entity