25/04/2023
Version 4

SPRING

DATA-JPA

SPRING DATA-JPA & HIBERNATE

PRÉSENTATION SPRING DATA-JPA

- Déclaration d'interfaces qui étendent l'interface `JpaRepository<T, ID>`
  - `@Repository` devient implicite
  - `@Transactional` devient implicite sur les méthodes classiques

PRÉSENTATION SPRING DATA-JPA

- Utilisation d'une convention pour nommer les méthodes
  - `findByNom(String nom)`
  - `findByReference(String reference)`
  - `findBy…(arguments attendus)`
  - `findByNomContaining(String nom)`

- Possibilité d'ajouter des `Queries` spécifiques
  - Utilisation de l'annotation `@Query("<Requête JP-QL>")` sur la méthode
  - Utilisation de l'annotation `@Modifying` sur la méthode
    - Si c'est une requête de mise à jour (`update` ou `delete`)

PRÉSENTATION SPRING DATA-JPA

```
public interface IProduitRepository extends JpaRepository<Produit, Integer> {
  public List<Produit> findByPrix(Double prix);
  public Produit findByNom(String nom);
  public List<Produit> findByNomContaining(String nom);

  @Query("select p from Produit p where p.libelle = ?1")
  public Produit findUnProduit(String libelleProduit);
}
```

- Si la méthode retourne un **Produit**
  - **Spring** retournera *null* si rien n'a été trouvé
  - **Spring** lèvera une exception si plusieurs éléments ont été trouvés
- Si la méthode retourne une liste de **Produit**
  - **Spring** retournera une liste !

PRÉSENTATION SPRING DATA-JPA

Convention	Effet
findByAttribut	Rechercher sur la valeur d'un attribut
findByAttribut1AndAttribut2	Rechercher sur la valeur de deux attributs
findByAttributContaining	Rechercher une valeur contenue dans un attribut
findAllByOrderByAttributDesc	Rechercher tout et ranger par ordre décroissant sur l'attribut
findFirstByAttribut	Rechercher le premier élément sur la valeur d'un attribut
findTop10ByAttribut	Rechercher les 10 premiers éléments sur la valeur d'un attribut
findTop10ByAttributContaining	Rechercher les 10 premiers éléments dont la valeur d'un attribut contient
findTop10ByAttributContainingOrderByIdDesc	Rechercher les 10 derniers éléments dont la valeur d'un attribut contient
countByLibelleContaining	Compter le nombre d'éléments dont la valeur d'un attribut contient

PRÉSENTATION SPRING DATA-JPA

```
public interface IProduitRepository extends JpaRepository<Produit, Integer> {
  public List<Produit> findByNom(String nom);
  public Produit findFirstByNom(String nom);
  public List<Produit> findTop10ByNomContainingOrderByIdDesc(String nom);
  public int countByNomContaining(String nom);
}
```

PRÉSENTATION SPRING DATA-JPA

- Important à savoir
  - Les méthodes `find` par défaut de **Spring DATA-JPA** retournent
    - Soit une `List<T>` (c'est le cas de *`findAll`*)
    - Soit un `Optional<T>` (c'est le cas de *`findById`*)
  - Si la convention n'est pas respectée et que la méthode n'est pas annotée de `@Query`
    - **Spring** génère une erreur au démarrage de l'application

PRÉSENTATION SPRING DATA-JPA

- Il est possible de paginer les résultats
  - En utilisant un objet de type **Pageable**

```
public List<Produit> findByNomContainingOrderByIdDesc(String nom, Pageable pageable);
```

```
// Première page de 20 éléments
repoProduit.findByNomContainingOrderByIdDesc("g", PageRequest.of(0, 20));

// Deuxième page de 20 éléments
repoProduit.findByNomContainingOrderByIdDesc("g", PageRequest.of(1, 20));
```

La pagination peut aller plus loin : possibilité de trier avec `PageRequest` !

CONFIGURATION

CONFIGURATION DE SPRING DATA-JPA

CONFIGURATION

- Une dépendance à charger (en plus des dépendances **SPRING JPA**)
  - **spring-data-jpa**

CONFIGURATION

- Annotation de la classe de configuration
  - `@EnableJpaRepositories("fr.formation.repo")`

EXERCICE

- Implémenter **SPRING DATA-JPA**