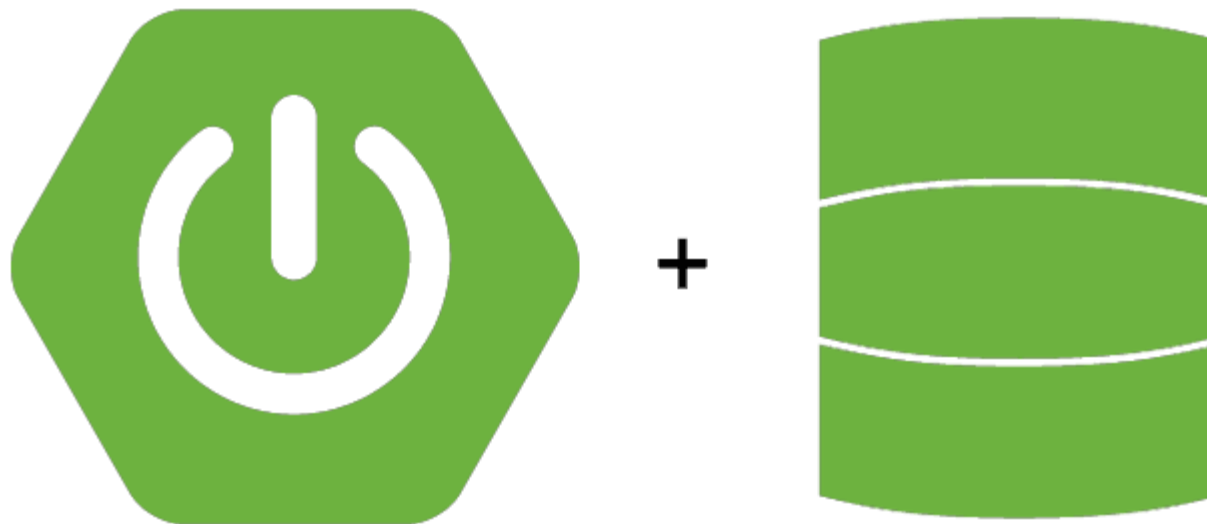


SPRING DATA JPA – Keywords



UP ASI
Bureaux E204 | E304

Plan du Cours

- Spring Data JPA + Keywords
- TP Spring Boot

SPRING DATA JPA

- Spring Data JPA va créer le code pour les méthodes que tu souhaites utiliser. Il suffit de lui indiquer les méthodes à utiliser :

Keyword	Sample	Equivalent to
GreaterThan	<code>findByDateDebutGreaterThan(Date dateC);</code>	Select p from ProjetDetail p where p.dateDebut > :dateC
LessThan	<code>findByDateDebutLessThan(Date dateN);</code>	Select p from ProjetDetail p where p.dateDebut < :dateN
Between	<code>findByDateDebutBetween(Date dFrom, Date dTo);</code>	Select p from ProjetDetail p where p.dateDebut between :dFrom and :dTo
IsNull, NotNull	<code>findByCout_provisoireNotNull();</code>	Select p from ProjetDetail p where p.cout_provisoire is not null
IsNull, Null	<code>findByDescriptionNull();</code>	Select p from ProjetDetail p where p.description is null
Like	<code>findByTechnologieLike(String technologie);</code>	Select p from ProjetDetail p where p. technologie like :technologie
(No keyword)	<code>findByTechnologie(String technologie);</code>	Select p from ProjetDetail p where p. technologie= :technologie
(No keyword)	<code>findOne(ID primaryKey);</code>	Select p from ProjetDetail p where p.id =:primaryKey
(No keyword)	<code>Long count();</code>	Select count(*) From Projet;

EXAMPLE

```
public interface ProjetDetailRepository
extends JpaRepository<ProjetDetail, Long>
{
```

```
// SELECT * FROM ProjetDetail WHERE technologie LIKE '%in%';
```

```
List<ProjetDetail> findByTechnologieLike(String technologie);
```

```
List<ProjetDetail> findByTechnologieContains(String technologie);
```

```
List<ProjetDetail> findByTechnologieContaining(String technologie);}
```

```
@Entity
@Table(name = "T_PROJET_DETAIL")
public class ProjetDetail implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "PD_ID")
    private Long id; // Identifiant projet detail (Clé p

    @Column(name = "PD_DESCRIPTION")
    private String description;

    @Column(name = "PD_TECHNOLOGIE")
    private String technologie;

    @Column(name = "PD_COUT_PROVISOIRE")
    private Long cout_provisoire;

    @Temporal(TemporalType.DATE)
    private Date dateDebut;

    @OneToOne(mappedBy = "projetDetail")
    private Projet projet;
}
```

EXEMPLE

- Afficher la liste des projets qui ont une technologie précise.

```
public interface ProjetRepository extends JpaRepository<Projet, Long> {  
    List<Projet> findByProjetDetailTechnologieContains(String technologie);  
}
```

association attribut dans la table Projet Detail

```
@Entity  
@Table(name = "T_PROJET")  
public class Projet implements Serializable {  
    private static final long serialVersionUID = 1L;  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "PROJET_ID")  
    private Long id; // Identifiant projet (Clé primaire)  
  
    @Column(name = "PROJET_SUJET")  
    private String sujet;  
  
    @OneToOne  
    private ProjetDetail projetDetail;  
}
```

EXEMPLE

- Afficher la liste des projets d'une équipe.

```
@Entity
@Table(name = "T_PROJET")
public class Projet implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "PROJET_ID")
    private Long id; // Identifiant projet (Clé primaire)

    @Column(name = "PROJET_SUJET")
    private String sujet;

    @ManyToMany(mappedBy = "projets", cascade = CascadeType.ALL)
    private Set<Equipe> equipes;
```

```
public interface ProjetRepository
extends JpaRepository<Projet, Long> {
    List<Projet> findByEquipesIdEquipe(Long equipeId);
}
```

EXEMPLE

- Afficher la liste des projets d'une équipe dont la description est non nulle.

```
public interface ProjetRepository extends JpaRepository<Projet, Long> {  
    List<Projet>  
    findByEquipesIdEquipeAndProjetDetailDescriptionNotNull(Long equipeId);  
}
```

EXEMPLE

```
public interface ProjetRepository extends JpaRepository<Projet, Long> {
```

- Afficher la liste des projets par équipe et entreprise.

```
List<Projet> findByEquipesIdEquipeAndEquipesEntrepriseIdEntreprise  
(Long equipeId, Long entrepriseId);
```

- Afficher la liste des projets par la spécialité d'une équipe et l'adresse de l'entreprise.

```
List<Projet>  
findByEquipesSpecialiteContainsAndEquipesEntrepriseAdresseContains  
(String specialite, String adresse);  
}
```


SPRING DATA JPA

Si vous avez des questions, n'hésitez pas à nous contacter :

Département Informatique
UP Architectures des Systèmes d'Information
Bureaux E204 | E304