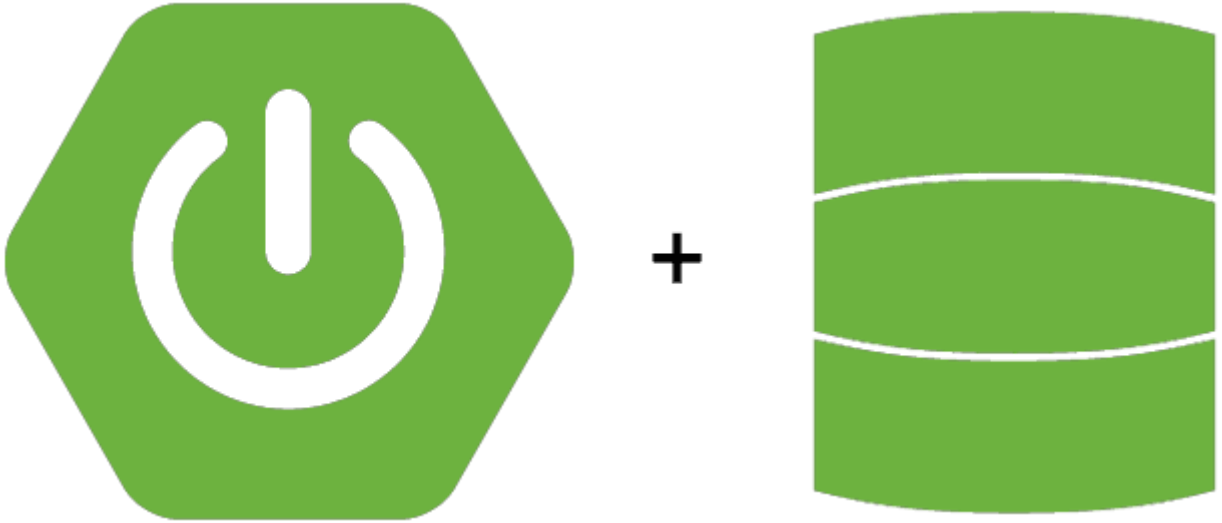
SPRING DATA JPA – CrudRepository

esprit
Se former autrement
HONORIS UNITED UNIVERSITIES



UP ASI
Bureau E204

Plan du Cours

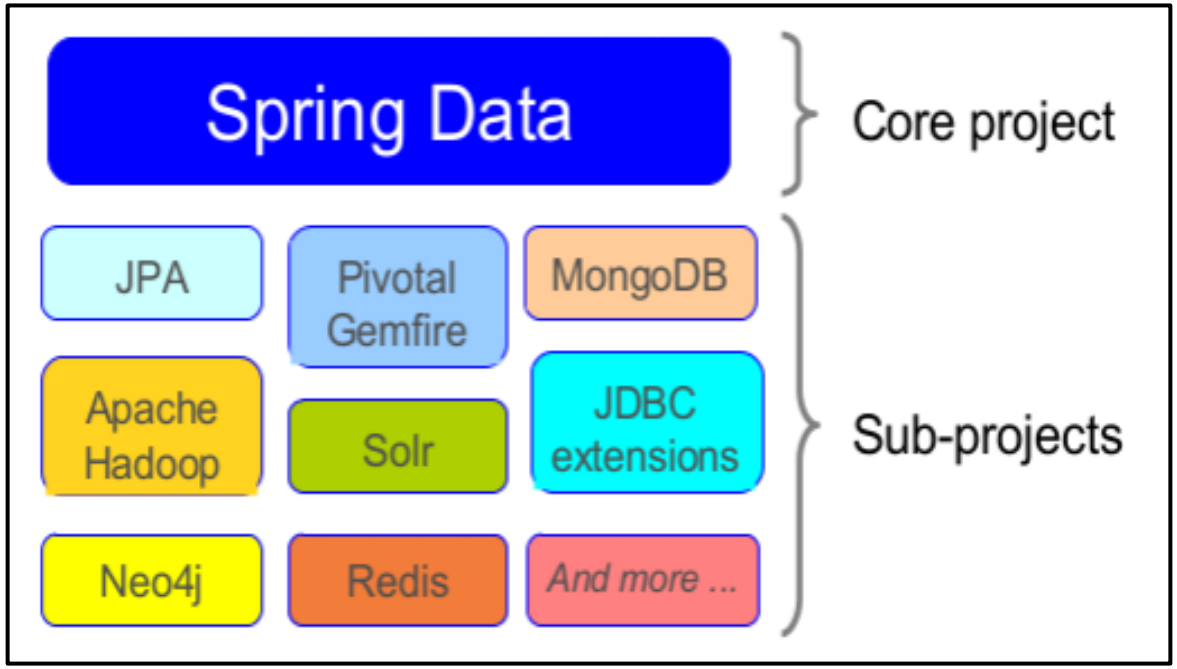
- Spring Data
- Spring Data JPA

- CRUD Repository interface
- Créer et Utiliser un Repository

- TP Spring Boot + Spring Data JPA

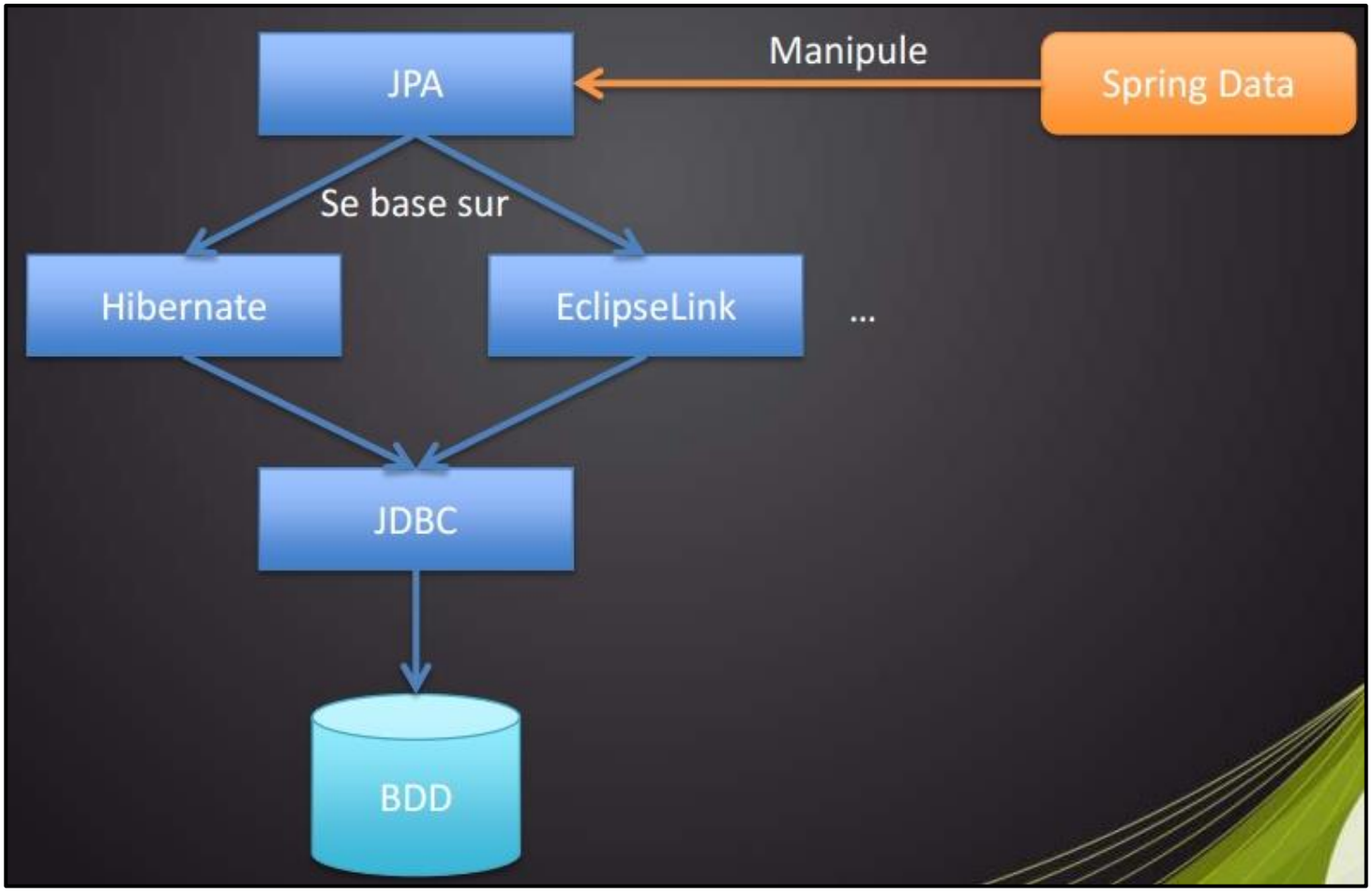
SPRING DATA

- C'est un module Spring qui a pour but de :
  - Faciliter l'écriture des couches d'accès aux données.
  - Offrir une abstraction commune pour l'accès aux données quelle que soit la source de données (SQL ou NoSQL).



SPRING DATA JPA

- Spring Data JPA est un sous projet du projet Spring Data.



SPRING DATA JPA

- Code répétitif avec les DAO (Data Access Object) avant Spring Data JPA:

```
public class ProjetDAO {

public Projet findById (Long id) {
//code ...
}

public List<Projet>findAll{
//code ...
}

}
```

```
public class EntrepriseDAO{

public Entreprise findById (Long id) {
//code ...
}

public List<Entreprise>findAll{
//code ...
}

}
```

SPRING DATA JPA

- Avec Spring Data JPA, il vous suffit de définir une interface qui permet de manipuler une entité, en étendant l'interface CrudRepository <T, ID> et de déclarer les méthodes pour manipuler cette entité.
- Spring Data JPA créera une classe qui implémente cette interface pour vous.
- Exemple :

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

- L'interface **ProjetRepository** étend l'interface **CrudRepository<Projet, Long>**. Elle contient les méthodes pour manipuler l'entité Projet.
- Spring Data JPA va automatiquement créer une classe qui implémente cette interface au moment de l'exécution de la l'application.

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

EXEMPLE

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

```
@Entity
@Table(name = "T_PROJET_DETAIL")
public class Projet_Detail 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;
}
```

```
// SELECT * FROM ProjetDetail WHERE technologie LIKE '%in%';
List<ProjetDetail> findByTechnologieLike(String technologie);
List<ProjetDetail> findByTechnologieContains(String technologie);
List<ProjetDetail> findByTechnologieContaining(String technologie);}
```

EXEMPLE

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

```
@Repository
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 primair

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

        @OneToOne
        private Projet_Detail projetDetail;

}
```

EXEMPLE

- Afficher la liste des projets d'une équipe.

```java
@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;
}
```

```java
@Repository
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.

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

EXEMPLE

```
@Repository
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);
}
```

Exercice

Dans l'étude de cas **Khaddem**, Réaliser les requetes suivantes en utilisant les keywords :

- Récupérer les étudiants d'un département donné
- Récupérer les étudiants dont l'équipe a un niveau SENIOR

SPRING DATA JPA (CrudRepository)

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

Département Informatique
UP Architectures des Systèmes d'Information
Bureau E204