

## Spring Data JPA

Guillaume Dufrêne – Lionel Seinturier

Université de Lille – Sciences et Technologies

## Définition

- interface d'accès aux données JPA
- générer de l'implémentation
- convention de nommage

## Exemple de code – Entity Personne

```
@Entity
public class Personne {

    private long id;
    private String nom;
    private int age;

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

    @Id
    @GeneratedValue
    public long getId() { return id; }
    public void setId( long id ) { this.id = id; }
    // méthodes getNom, setNom, getAge, setAge
}
```

- Interface des opérations CRUD, **définie par Spring**

```
public interface CrudRepository<T, ID extends Serializable>
    extends Repository<T, ID> {

    <S extends T> S save( S entity );
    T findOne( ID id );
    void delete( ID id );

    long count();
    boolean exists( ID primaryKey );
    void delete( T entity );
    void deleteAll();
    Iterable<T> findAll();
    Iterable<T> findAll( Iterable<ID> ids );
    <S extends T> Iterable<S> save( S entity );
}
```

## Exemple de code – Repository Personne

```
public interface PersonneRepository
extends CrudRepository<Personne, Long> {}

@Controller
public class PersonneController {

    @Autowired
    private PersonneRepository pr;

    @GetMapping("/personne/{id}")
    public void one( @PathVariable long id, Model model ) {
        Personne p = pr.findOne(id);
        model.addAttribute("personne", p); }

    @GetMapping("/personnes")
    public void all( Model model ) {
        Iterable<Person> all = pr.findAll();
        model.addAtttribute("liste", all); } }
```

- définition de requêtes propres au domaine de l'Entity
- convention de nommage pour les méthodes
  - find ... By *attribut*

```
public interface PersonneRepository
extends CrudRepository<Personne, Long> {

    List<Personne> findByNom( String nom );
    List<Personne> findByNomAndAge( String nom, int age );
    List<Personne> findByNomIgnoreCase( String nom );
    List<Personne> findByNomOrderByAgeAsc( String nom );
    List<Personne> findByNomOrderByAgeDesc( String nom );
}
```

And, Or,  
LessThan,  
GreaterThan  
Like, Between

- les résultats peuvent être retournés par lots (pages)
  - type **Pageable**

```
public interface PersonneRepository
extends CrudRepository<Personne, Long> {

    List<Personne> findAll( Pageable p );
    List<Personne> findByNom( String nom, Pageable p );
}
```

## Invocation

```
@Autowired
private PersonneRepository pr;

Page<Personnes> personnes = pr.findAll( new PageRequest(1, 20) );
personnes = pr.findByNom( "Bob", new PageRequest(1, 5) );
```

## Spring Data

- les données manipulées par l'application
- une mise en oeuvre pour JPA
- interface repository
- convention de nommage pour l'écriture des requêtes