everis

an NTT DATA Company

BECA Java

everis
an NTT DATA Company

Agenda

- Spring Data
  - Introducción
  - Repositories
  - Tipos de consulta

everis
an NTT DATA Company

01

Spring Data JPA

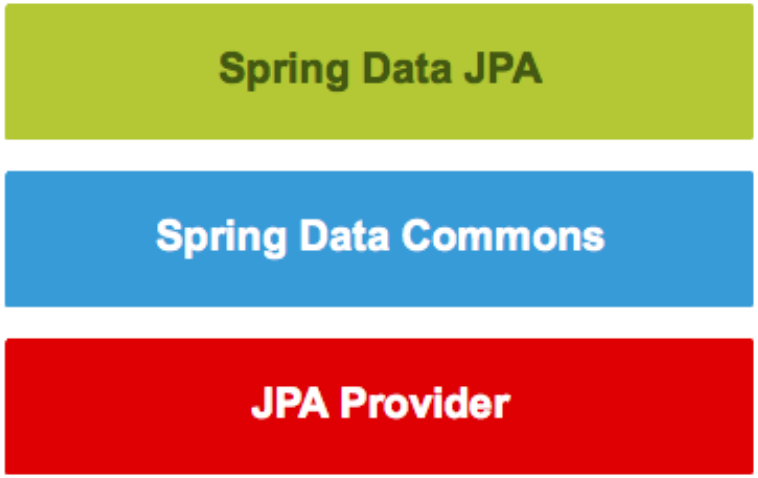
everis
an NTT DATA Company

Introducción

- Spring Data no es un proveedor/implementación de JPA
- Es una librería que añade un nivel de abstracción adicional sobre un proveedor de JPA.

Introducción

- Si decidimos añadir Spring Data JPA a nuestro proyecto, tenemos que tener en cuenta estas tres capas
  - Spring Data JPA: Soporte para crear repositororios extendiendo las interfaces Repository
  - Commons: Provee la insfraestructura compartida por los diferentes proyectos spring data
  - JPA provider: hibernate?



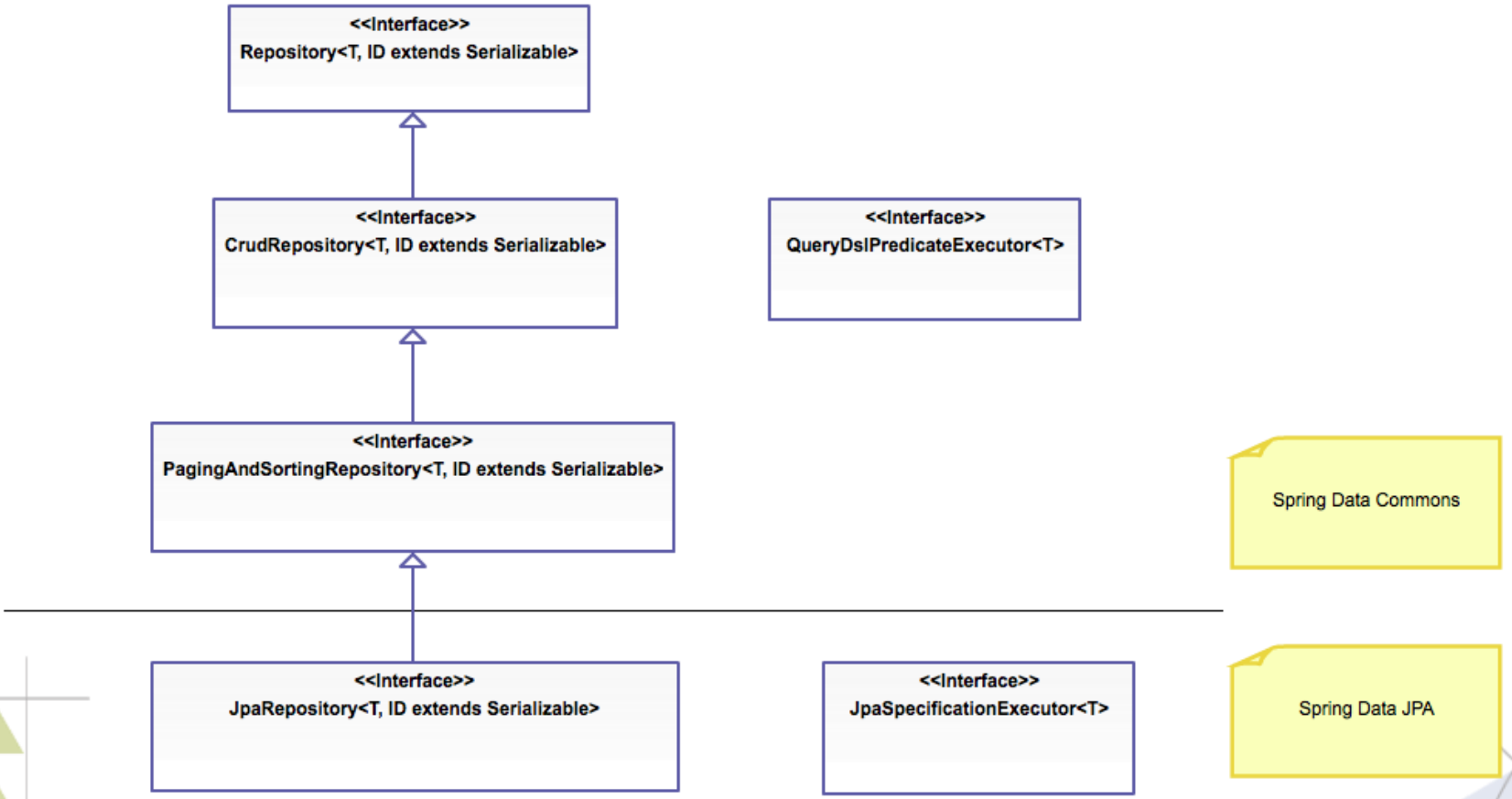
everis
an NTT DATA Company

Spring Data Repositories

- Commons
  - ***Repository<T, ID extends Serializable>***
  - ***CrudRepository<T, ID extends Serializable>***
  - ***PagingAndSortingRepository<T, ID extends Serializable>***
  - ***QueryDslPredicateExecutor<T>:*** nos permite hacer consultas más complejas con http://www.querydsl.com
- Específicas JPA
  - ***JpaRepository<T, ID extends Serializable>***
  - ***JpaSpecificationExecutor<T>***

everis
an NTT DATA Company

Spring Data Repositories



everis
an NTT DATA Company

Spring Data Repositories

- Cómo se usan?
  - Crea una interfaz que extienda una de esas interfaces
  - (Opcionalmente) añade métodos customizados para tus entidades
  - Injecta la interfaz dentro de un Bean de spring

everis
an NTT DATA Company

Spring Data Repositories

- Qué tipos de queries tenemos?
  - Por nombres de método*
  - Con @Query*
  - Con NamedQueries*
  - Criteria
  - Specifications
  - QueryDsl

everis
an NTT DATA Company

Spring Data Repositories – Nombre de método

- https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#jpa.query-methods.query-creation

Keyword	Sample	JPQL snippet
And	findByLastnameAndFirstname	… where x.lastname = ?1 and x.firstname = ?2
Or	findByLastnameOrFirstname	… where x.lastname = ?1 or x.firstname = ?2
Is,Equals	findByFirstname,findByFirstnameIs,findByFirstnameEquals	… where x.firstname = ?1
Between	findByStartDateBetween	… where x.startDate between ?1 and ?2
LessThan	findByAgeLessThan	… where x.age < ?1
LessThanEqual	findByAgeLessThanEqual	… where x.age <= ?1
GreaterThan	findByAgeGreaterThan	… where x.age > ?1
GreaterThanEqual	findByAgeGreaterThanEqual	… where x.age >= ?1
After	findByStartDateAfter	… where x.startDate > ?1
Before	findByStartDateBefore	… where x.startDate < ?1
IsNull	findByAgeIsNull	… where x.age is null
IsNotNull,NotNull	findBy(Is)NotNull	… where x.age not null
Like	findByFirstnameLike	… where x.firstname like ?1

everis
an NTT DATA Company

Spring Data Repositories – @Query

- Nos permite ejecutar consular JPQL/HQL o nativas anotando el método con @Query y definiendo la query dentro

```
@Query("select u from User u where u.emailAddress = ?1")
User findByEmailAddress(String emailAddress);
```

```
@Query(value = "SELECT * FROM USERS WHERE EMAIL_ADDRESS = ?1", nativeQuery = true)
User findByEmailAddress(String emailAddress);
```

```
@Query("select u from User u where u.firstname = :firstname or u.lastname = :lastname")
User findByLastnameOrFirstname(@Param("lastname") String lastname,
                               @Param("firstname") String firstname);
}
```

```
@Query("select u from #{#entityName} u where u.lastname = ?1")
List<User> findByLastname(String lastname);
```

everis
an NTT DATA Company

Spring Data Repositories – NamedQuery

- Nombramos la named query siguiendo el patrón @{EntityName}.{queryName}

```
@Entity
@Table(name = "employee", schema="spring_data_jpa_example")
@NamedQuery(name = "Employee.fetchByLastNameLength",
        query = "SELECT e FROM Employee e WHERE CHAR_LENGTH(e.lastname) =:length "
)
public class Employee {
```

- Declaramos un método que coincida con el queryName pasándole los argumentos con @Param

```
@Repository
public interface EmployeeRepository extends JpaRepository<Employee,Long>, EmployeeRepositoryCustom {

    List<Employee> fetchByLastNameLength(@Param("length") Long length);
}
```