

Spring Data Repository per le operazioni di CRUD - Parte I

Cosa imparerai

- Cos'è Spring Data Repository e come utilizzarlo al meglio per le operazioni di Create-Read-Update-Delete

Spring Data Repository

Spring Data ha un potente meccanismo per la gestione dei metodi di accesso al Database in lettura e scrittura.

Questo **livello di astrazione** è implementato attraverso una serie di strumenti che consentono di automatizzare la creazione dei metodi di gestione dei dati (creazione di una riga, ricerca di elementi sul DB, ecc...)

In Spring Data ci sono **due elementi alla base di questo processo** e sono **due interfacce**:

- ❑ **Repository**, svolge il ruolo di marker e serve per identificare le interfacce delegate alla gestione dell'accesso ai dati
- ❑ **CrudRepository**, definisce le principali operazioni di CRUD

Spring Data Repository

L'interfaccia CrudRepository

```
public interface CrudRepository<T, ID> extends Repository<T, ID> {  
  
    <S extends T> S save(S entity);           1  
  
    Optional<T> findById(ID primaryKey);      2  
  
    Iterable<T> findAll();                    3  
  
    long count();                             4  
  
    void delete(T entity);                     5  
  
    boolean existsById(ID primaryKey);        6  
  
    // ... more functionality omitted.  
}
```

- 1 Saves the given entity.
- 2 Returns the entity identified by the given ID.
- 3 Returns all entities.
- 4 Returns the number of entities.
- 5 Deletes the given entity.
- 6 Indicates whether an entity with the given ID exists.

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#repositories.core-concepts>

Spring Data Repository

Query Method

Uno degli elementi più interessanti presenti in Spring Data è il *Query Method*.

Spring Data consente, infatti, di definire dei metodi (tipicamente quelli di ricerca personalizzata...) in un'interfaccia che estenderà *CrudRepository* utilizzando una sintassi predefinita.

L'implementazione del metodo definito nell'interfaccia verrà fatta in automatico da Spring.

Spring Data Repository

Come faccio ad utilizzare Spring Data Repository?

1. Creo un'interfaccia che estende CrudRepository
2. Definisco i metodi nell'interfaccia
3. Abilito l'utilizzo dei Repository inserendo l'annotation `@EnableJpaRepositories` nella classe di configurazione (o tramite l'elemento `<jpa:repositories base-package="..."/>` se usiamo la notazione XML)

Spring Data Repository

Qual è la sintassi da utilizzare per la definizione dei metodi nell'interfaccia?

Spring Data mette a disposizione una serie di parole chiave che possono essere utilizzate per comporre nomi di metodi anche complessi.

Ad esempio:

- ❑ **find...By, get...By, search...By** sono combinazioni di parole chiave usate da Spring Data per implementare metodi di ricerca(**findContactByName**(String name), **getContactById**(Long id), ...)
- ❑ **count...By** è una combinazione utilizzata per definire metodi che restituiscono il numero di elementi presenti in una tabella (**countContactByName**(String name),...)

Spring Data Repository

Qual è la sintassi da utilizzare per la definizione dei metodi nell'interfaccia?

Possiamo anche definire metodi per effettuare query più complesse (ad es. una join tra parametri...)

Alcune parole chiave che possiamo utilizzare sono:

- ❑ **And** (ad es. `findByNameAndCity(String name, String city), ...`)
- ❑ **Or** (ad es. `findByNameOrCity(String name, String city), ...`)
- ❑ **In** (ad es. `findByNameIn(Collection<String> names), ...`)
- ❑ **Distinct** (ad es. `findDistinctByName(String name), ...`)
- ❑ **GreaterThan, GreaterThanEqual** (ad es. `findByIdGreaterThan(Long id), ...`)
- ❑ **StartingWith** (ad es. `findUserByNameStartingWith(String name), ...`)
- ❑ **Like, NotLike** (ad es. `findUserByNameLike(String name), ...`)
- ❑ ...

Tutte le keywords supportate le trovate sulla documentazione ufficiale!

<https://docs.spring.io/spring-data/jpa/docs/current/reference/html/#repository-query-keywords>

Di cosa abbiamo parlato in questa lezione

- Cos'è Spring Data Repository e come utilizzarlo al meglio per le operazioni di Create-Read-Update-Delete