

Spring Data JPA Methods

By Ramesh Fadatare (Java Guides)


```
@NoRepositoryBean
public interface PagingAndSortingRepository<T, ID> extends CrudRepository<T, ID> {
    Iterable<T> findAll(Sort sort);

    Page<T> findAll(Pageable pageable);
}
```

```
@NoRepositoryBean
public interface JpaRepository<T, ID> extends PagingAndSortingRepository<T, ID> {
    List<T> findAll();

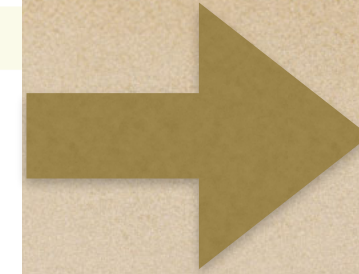
    List<T> findAll(Sort sort);
    List<T> findAllById(Iterable<ID> ids);
    <S extends T> List<S> saveAll(Iterable<S> entities);
    void flush();
    <S extends T> S saveAndFlush(S entity);
    <S extends T> List<S> saveAllAndFlush(Iterable<S> entities);

    @Deprecated
    @Deprecated
    default void deleteInBatch(Iterable<T> entities) { this.deleteAllInBatch(entities); }
    void deleteAllInBatch(Iterable<T> entities);
    void deleteAllByIdInBatch(Iterable<ID> ids);
    void deleteAllInBatch();

    @Deprecated
    @Deprecated
    T getOne(ID id);
    T getById(ID id);

    <S extends T> List<S> findAll(Example<S> example);
    <S extends T> List<S> findAll(Example<S> example, Sort sort);
}
```

```
public interface ProductRepository extends JpaRepository<Product, Long> {
    |
}
```



```
@NoRepositoryBean
public interface CrudRepository<T, ID> extends Repository<T, ID> {
    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

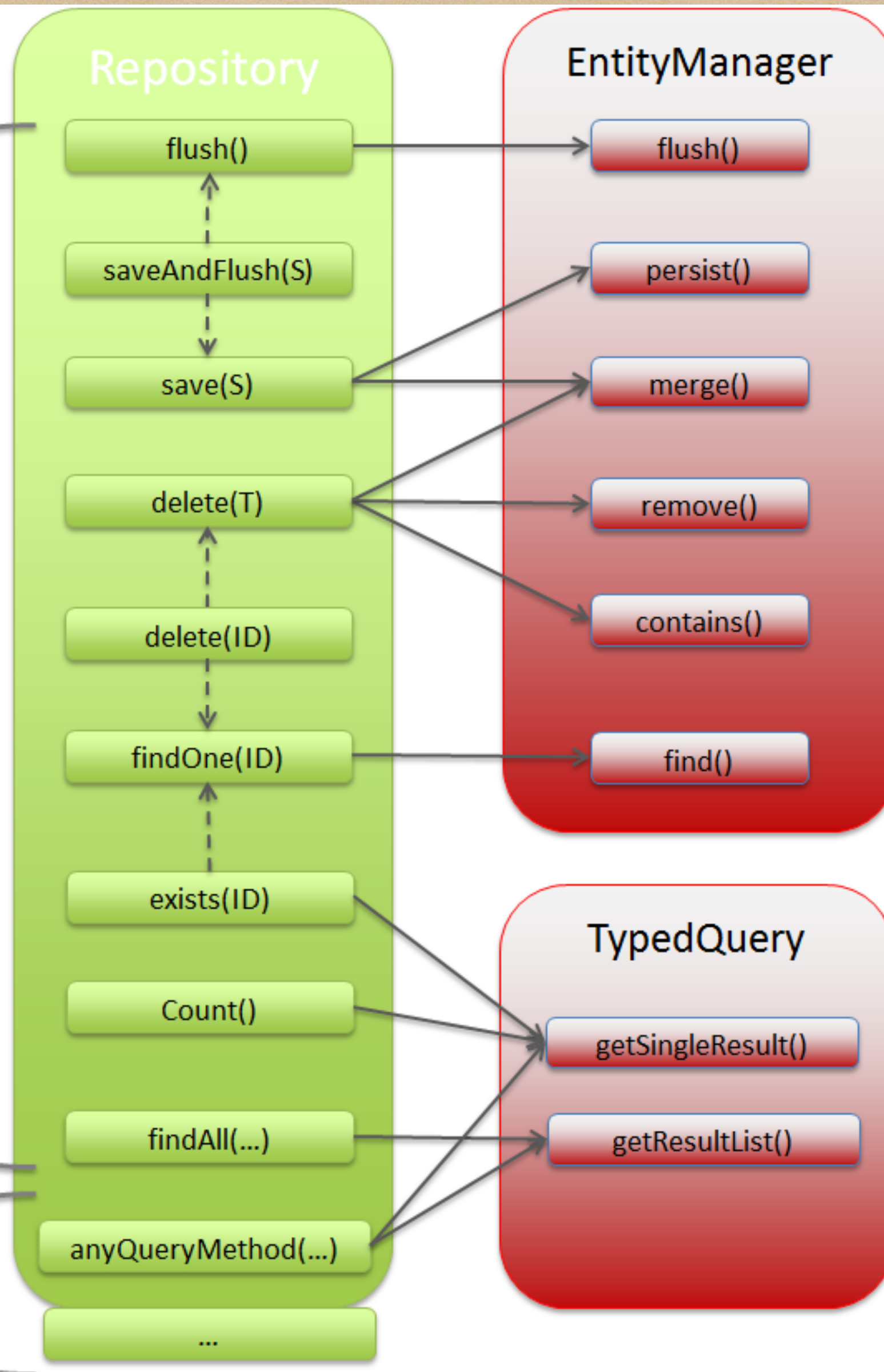
    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();
}
```


Methods of
JpaRepository
Interface.

Query Methods in
Repository Interface for
Entity that defined by
developer.



save() method

As the name depicts, the save() method allows us to save an entity to the DB.

Saving an entity can be performed with the `CrudRepository.save(...)` method. It persists or merges the given entity by using the underlying JPA **EntityManager**. If the entity has not yet been persisted, Spring Data JPA saves the entity with a call to the `entityManager.persist(...)` method. Otherwise, it calls the `entityManager.merge(...)` method.

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();
}
```


Update existing entity using save() method

As the name depicts, the save() method allows us to save an entity to the DB.

Saving an entity can be performed with the CrudRepository.save(...) method. It persists or merges the given entity by using the underlying JPA **EntityManager**. If the entity has not yet been persisted, Spring Data JPA saves the entity with a call to the entityManager.persist(...) method. Otherwise, it calls the entityManager.merge(...) method.

```
@NoRepositoryBean
public interface CrudRepository<T, ID> extends Repository<T, ID> {
    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();
}
```


saveAll() method

As the name depicts, the saveAll() method allows us to save multiple entities to the DB.

It belongs to the CrudRepository interface defined by Spring Data.

saveAll() method returns a list of Iterable objects

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();

}
```


findById() method

As the name depicts, the findById() method allows us to get or retrieve an entity based on a given id (primary key) from the DB.

It belongs to the CrudRepository interface defined by Spring Data.

findById() method returns Optional of type Entity

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();

}
```


findAll() method

As the name depicts, the findAll() method allows us to get or retrieve all the entities from the database table.

It belongs to the CrudRepository interface defined by Spring Data.

findById() method returns List of Iterable objects

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();

}
```


deleteById() method

As the name depicts, the deleteById() method allows us to delete an entity by id from the database table.

It belongs to the CrudRepository interface defined by Spring Data.

deleteById() method returns void (nothing)

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();

}
```


delete() method

As the name depicts, the delete() method allows us to delete an entity from the database table.

It belongs to the CrudRepository interface defined by Spring Data.

delete() method returns void (nothing)

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();
}
```


deleteAll() two overloaded methods

As the name depicts, the deleteAll() method allows us to delete all the entities from the database table.

It belongs to the CrudRepository interface defined by Spring Data.

deleteAll() method returns void (nothing)

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();

}
```


existsById() method

As the name depicts, the existsById() method allows us to check if the entity exists with a given id in a database table.

It belongs to the CrudRepository interface defined by Spring Data.

existsById() method returns boolean (true or false)

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

    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();
}
```


count() method

As the name depicts, the count() method allows us to count the number of records that exist in a database table.

It belongs to the CrudRepository interface defined by Spring Data.

count() method returns long (numbers of records)

```
@NoRepositoryBean
public interface CrudRepository<T, ID> extends Repository<T, ID> {
    <S extends T> S save(S entity);

    <S extends T> Iterable<S> saveAll(Iterable<S> entities);

    Optional<T> findById(ID id);

    boolean existsById(ID id);

    Iterable<T> findAll();

    Iterable<T> findAllById(Iterable<ID> ids);

    long count();

    void deleteById(ID id);

    void delete(T entity);

    void deleteAllById(Iterable<? extends ID> ids);

    void deleteAll(Iterable<? extends T> entities);

    void deleteAll();
}
```