Package org.springframework.data.repository

Interface CrudRepository<T,ID>

All Superinterfaces:

`Repository<T,ID>`

All Known Subinterfaces:

`ListCrudRepository<T,ID>`

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

Interface for generic CRUD operations on a repository for a specific type.

Author:

Oliver Gierke, Eberhard Wolff, Jens Schauder

Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type	Method	Description
`long`	`count()`	Returns the number of entities available.
`void`	`delete(T entity)`	Deletes a given entity.
`void`	`deleteAll()`	Deletes all entities managed by the repository.
`void`	`deleteAll(Iterable <? extends T> entities)`	Deletes the given entities.
`void`	`deleteAllById(Iterable <? extends ID> ids)`	Deletes all instances of the type T with the given IDs.
`void`	`deleteById(ID id)`	Deletes the entity with the given id.
`boolean`	`existsById(ID id)`	Returns whether an entity with the given id exists.
`Iterable <T>`	`findAll()`	Returns all instances of the type.
`Iterable <T>`	`findAllById(Iterable <ID> ids)`	Returns all instances of the type T with the given IDs.
`Optional <T>`	`findById(ID id)`	Retrieves an entity by its id.
`<S extends T> S`	`save(S entity)`	Saves a given entity.
`<S extends T> Iterable <S>`	`saveAll(Iterable <S> entities)`	Saves all given entities.

Method Details

save

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

Saves a given entity. Use the returned instance for further operations as the save operation might have changed the entity instance completely.

Parameters:

`entity` - must not be null.

Returns:

the saved entity; will never be null.

Throws:

`IllegalArgumentException` - in case the given entity is null.

OptimisticLockingFailureException - when the entity uses optimistic locking and has a version attribute with a different value from that found in the persistence store. Also thrown if the entity is assumed to be present but does not exist in the database.

saveAll

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

Saves all given entities.

Parameters:

`entities` - must not be null nor must it contain null.

Returns:

the saved entities; will never be null. The returned Iterable will have the same size as the Iterable passed as an argument.

Throws:

`IllegalArgumentException` - in case the given `entities` or one of its entities is null.

`OptimisticLockingFailureException` - when at least one entity uses optimistic locking and has a version attribute with a different value from that found in the persistence store. Also thrown if at least one entity is assumed to be present but does not exist in the database.

findById

```
Optional <T> findById(ID id)
```

Retrieves an entity by its id.

Parameters:

`id` - must not be null.

Returns:

the entity with the given id or Optional#empty() if none found.

Throws:

`IllegalArgumentException` - if id is null.

existsById

```
boolean existsById(ID id)
```

Returns whether an entity with the given id exists.

Parameters:

`id` - must not be null.

Returns:

true if an entity with the given id exists, false otherwise.

Throws:

`IllegalArgumentException` - if id is null.

findAll

```
Iterable <T> findAll()
```

Returns all instances of the type.

Returns:

all entities

findAllById

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

Returns all instances of the type `T` with the given IDs.

If some or all ids are not found, no entities are returned for these IDs.

Note that the order of elements in the result is not guaranteed.

Parameters:

`ids` - must not be null nor contain any null values.

Returns:

guaranteed to be not null. The size can be equal or less than the number of given ids.

Throws:

`IllegalArgumentException` - in case the given `ids` or one of its items is null.

count

```
long count()
```

Returns the number of entities available.

Returns:

the number of entities.

deleteById

```
void deleteById(ID id)
```

Deletes the entity with the given id.

If the entity is not found in the persistence store it is silently ignored.

Parameters:

`id` - must not be null.

Throws:

`IllegalArgumentException` - in case the given id is null

delete

```
void delete(T entity)
```

Deletes a given entity.

Parameters:

`entity` - must not be null.

Throws:

`IllegalArgumentException` - in case the given entity is null.

`OptimisticLockingFailureException` - when the entity uses optimistic locking and has a version attribute with a different value from that found in the persistence store. Also thrown if the entity is assumed to be present but does not exist in the database.

deleteAllById

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

Deletes all instances of the type `T` with the given IDs.

Entities that aren't found in the persistence store are silently ignored.

Parameters:

`ids` - must not be null. Must not contain null elements.

Throws:

`IllegalArgumentException` - in case the given ids or one of its elements is null.

Since:

2.5

deleteAll

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

Deletes the given entities.

Parameters:

`entities` - must not be null. Must not contain null elements.

Throws:

`IllegalArgumentException` - in case the given entities or one of its entities is null.

`OptimisticLockingFailureException` - when at least one entity uses optimistic locking and has a version attribute with a different value from that found in the persistence store. Also thrown if at least one entity is assumed to be present but does not exist in the database.

deleteAll

```
void deleteAll()
```

Deletes all entities managed by the repository.