

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)[org.springframework.data.repository](#)

Interface CrudRepository<T,ID>

All Superinterfaces:

[Repository<T, ID>](#)

All Known Subinterfaces:

[PagingAndSortingRepository<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

Method Summary

All Methods **Instance Methods** **Abstract Methods**

Modifier and Type	Method and 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	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 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 Detail

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.

saveAll

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

Saves all given entities.

Parameters:

entities - must not be null.

Returns:

the saved entities will never be null.

Throws:

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

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 with the given IDs.

Parameters:

`ids` -

Returns:**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.

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 -

Throws:

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

deleteAll

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

Deletes the given entities.

Parameters:

entities -

Throws:

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

deleteAll

```
void deleteAll()
```

Deletes all entities managed by the repository.

[PREV CLASS](#) **[NEXT CLASS](#)** [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

Copyright © 2011–2017 Pivotal Software, Inc.. All rights reserved.

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)`org.springframework.data.repository`

Interface `Repository<T,ID>`

Type Parameters:

`T` - the domain type the repository manages

`ID` - the type of the id of the entity the repository manages

All Known Subinterfaces:

`CrudRepository<T,ID>`, `PagingAndSortingRepository<T,ID>`,
`ReactiveCrudRepository<T,ID>`, `ReactiveSortingRepository<T,ID>`,
`RevisionRepository<T,ID,N>`, `RxJava2CrudRepository<T,ID>`,
`RxJava2SortingRepository<T,ID>`

`@Indexed`

public interface **`Repository<T,ID>`**

Central repository marker interface. Captures the domain type to manage as well as the domain type's id type. General purpose is to hold type information as well as being able to discover interfaces that extend this one during classpath scanning for easy Spring bean creation.

Domain repositories extending this interface can selectively expose CRUD methods by simply declaring methods of the same signature as those declared in `CrudRepository`.

Author:

Oliver Gierke

See Also:

`CrudRepository`

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) [DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

Copyright © 2011-2017 Pivotal Software, Inc.. All rights reserved.

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)[SUMMARY: FIELD](#) | [REQUIRED](#) | [OPTIONAL](#) [DETAIL: FIELD](#) | [ELEMENT](#)`org.springframework.data.repository`

Annotation Type RepositoryDefinition

`@Indexed``@Documented``@Retention(value=RUNTIME)``@Target(value=TYPE)``@Inherited``public @interface RepositoryDefinition`

Annotation to demarcate interfaces a repository proxy shall be created for. Annotating an interface with `RepositoryDefinition` will cause the same behaviour as extending `Repository`.

Author:`Oliver Gierke`**See Also:**`Repository`

Required Element Summary

Required Elements

Modifier and Type	Required Element and Description
<code>Class<?></code>	domainClass The domain class the repository manages.
<code>Class<?></code>	idClass The id class of the entity the repository manages.

Element Detail

domainClass

`public abstract Class<?> domainClass`

The domain class the repository manages. Equivalent to the T type parameter in `Repository`.

Returns:**See Also:**

Repository

idClass

```
public abstract Class<?> idClass
```

The id class of the entity the repository manages. Equivalent to the ID type parameter in [Repository](#).

Returns:

See Also:

[Repository](#)

Spring Data Core

[OVERVIEW](#) [PACKAGE](#) [CLASS](#) [USE](#) [TREE](#) [DEPRECATED](#) [INDEX](#) [HELP](#)

PREV CLASS [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [ALL CLASSES](#)

SUMMARY: [FIELD](#) | [REQUIRED](#) | [OPTIONAL](#) [DETAIL: FIELD](#) | [ELEMENT](#)

Copyright © 2011–2017 Pivotal Software, Inc.. All rights reserved.