Package org.springframework.data.domain

Class Sort

java.lang.Object
    org.springframework.data.domain.Sort

All Implemented Interfaces:

`Serializable` , `Iterable` <`Sort.Order`>, `Supplier` <`Stream` <`Sort.Order`>>, `Streamable<Sort.Order>`

Direct Known Subclasses:

`QSort`, `RevisionSort`, `Sort.TypedSort`

```
public class Sort
extends Object
implements Streamable<Sort.Order>, Serializable
```

Sort option for queries. You have to provide at least a list of properties to sort for that must not include null or empty strings. The direction defaults to `DEFAULT_DIRECTION`.

Author:

Oliver Gierke, Thomas Darimont, Mark Paluch, Johannes Englmeier

See Also:

Serialized Form

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
`static enum`	**`Sort.Direction`**	Enumeration for sort directions.
`static enum`	**`Sort.NullHandling`**	Enumeration for null handling hints that can be used in `Sort.Order` expressions.
`static class`	**`Sort.Order`**	PropertyPath implements the pairing of an `Sort.Direction` and a property.
`static class`	**`Sort.TypedSort`**`<T>`	Extension of Sort to use method handles to define properties to sort by.

Field Summary

Fields

Modifier and Type	Field	Description
`static final Sort.Direction`	`DEFAULT_DIRECTION`	

Constructor Summary

Constructors

Modifier	Constructor	Description
`protected`	`Sort(List <Sort.Order> orders)`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
Sort	and(Sort sort)	Returns a new Sort consisting of the Sort.Orders of the current Sort combined with the given ones.
Sort	ascending()	Returns a new Sort with the current setup but ascending order direction.
static Sort	by(String ... properties)	Creates a new Sort for the given properties.
static Sort	by(List <Sort.Order> orders)	Creates a new Sort for the given Sort.Orders.
static Sort	by (Sort.Direction direction, String ... properties)	Creates a new Sort for the given Sort.Direction and properties.
static Sort	by(Sort.Order... orders)	Creates a new Sort for the given Sort.Orders.
Sort	descending()	Returns a new Sort with the current setup but descending order direction.
protected List <Sort.Order>	doReverse()	
boolean	equals(Object obj)	
Sort.Order	getOrderFor(String property)	Returns the order registered for the given property.
int	hashCode()	
boolean	isEmpty()	Returns whether the current Streamable is empty.
boolean	isSorted()	
boolean	isUnsorted()	
Iterator <Sort.Order>	iterator()	
Sort	reverse()	Returns a new Sort with reversed sort Sort.Orders turning effectively asccending into descending sort order and vice versa.

static `<T> Sort.TypedSort<T>`	`sort(Class <T> type)`	Creates a new `Sort.TypedSort` for the given type.
`String`	`toString()`	
static `Sort`	`unsorted()`	Returns a `Sort` instances representing no sorting setup at all.

Methods inherited from class java.lang.Object

`clone , finalize , getClass , notify , notifyAll , wait , wait , wait`

Methods inherited from interface java.lang.Iterable

`forEach , spliterator`

Methods inherited from interface org.springframework.data.util.Streamable

`and, and, and, and, filter, flatMap, get, map, stream, toList, toSet`

Field Details

DEFAULT_DIRECTION

```
public static final Sort.Direction DEFAULT_DIRECTION
```

Constructor Details

Sort

```
protected Sort(List <Sort.Order> orders)
```

Method Details

by

```
public static Sort by(String ... properties)
```

Creates a new Sort for the given properties.

Parameters:

`properties` - must not be null.

Returns:

by

```
public static Sort by(List <Sort.Order> orders)
```

Creates a new Sort for the given Sort.Orders.

Parameters:

`orders` - must not be null.

Returns:

by

```
public static Sort by(Sort.Order... orders)
```

Creates a new Sort for the given Sort.Orders.

Parameters:

`orders` - must not be null.

Returns:

by

```
public static Sort by(Sort.Direction direction,
                      String ... properties)
```

Creates a new Sort for the given Sort.Direction and properties.

Parameters:

`direction` - must not be null.

`properties` - must not be null.

Returns:

sort

```
public static <T> Sort.TypedSort<T> sort(Class <T> type)
```

Creates a new Sort.TypedSort for the given type.

Parameters:

`type` - must not be null.

Returns:

Since:

2.2

unsorted

```
public static Sort unsorted()
```

Returns a `Sort` instances representing no sorting setup at all.

Returns:

descending

```
public Sort descending()
```

Returns a new `Sort` with the current setup but descending order direction.

Returns:

ascending

```
public Sort ascending()
```

Returns a new `Sort` with the current setup but ascending order direction.

Returns:

isSorted

```
public boolean isSorted()
```

isEmpty

```
public boolean isEmpty()
```

Description copied from interface: `Streamable`

Returns whether the current `Streamable` is empty.

Specified by:

`isEmpty` in interface `Streamable`<`Sort.Order`>

Returns:

isUnsorted

```
public boolean isUnsorted()
```

and

```
public Sort and(Sort sort)
```

Returns a new `Sort` consisting of the `Sort.Order`s of the current `Sort` combined with the given ones.

Parameters:

`sort` - must not be null.

Returns:

reverse

```
public Sort reverse()
```

Returns a new `Sort` with reversed sort `Sort.Order`s turning effectively asccending into descending sort order and vice versa.

Returns:

a new `Sort` object with reversed sort orders applied.

Since:

3.1

doReverse

```
protected List <Sort.Order> doReverse()
```

getOrderFor

```
@Nullable
public Sort.Order getOrderFor(String  property)
```

Returns the order registered for the given property.

Parameters:

`property` -

Returns:

iterator

```
public Iterator <Sort.Order> iterator()
```

Specified by:

`iterator` in interface `Iterable <Sort.Order>`

equals

```
public boolean equals(@Nullable
                      Object  obj)
```

Overrides:

`equals` in class `Object`

hashCode

```
public int hashCode()
```

Overrides:

`hashCode` in class `Object`

toString

```
public String  toString()
```

Overrides:

`toString` in class `Object`