

# WONder-ERXArrayUtilities

## Overview

ERXArrayUtilities provides convenience methods and tools for manipulating NSArray. Many of the methods are self explanatory. For a full list please see the api (<http://wocommunity.org/documents/javadoc/wonder/latest/extensions/foundation/ERXArrayUtilities.html>).

## NSArray operators

WebObjects provides @sum, @avg, etc array operators. ERXArrayUtilities adds quite a few more:

**SortOperator:** Define an NSArray.Operator for the key *sort*.

This allows for key value paths like:

```
myArray.valueForKey("@sort.firstName");
```

```
myArray.valueForKey("@sort.lastName,firstName");
```

Which in the first case would return myArray sorted ascending by first name and the second case by lastName and then by firstName.

Other sort operators registered are: @sortAsc, @sortDesc, @sortInsensitiveAsc, @sortInsensitiveDesc

**FetchSpecOperator:** Define an NSArray.Operator for the key *fetchSpec*.

This allows for key value paths like:

```
myArray.valueForKey("@fetchSpec.fetchUsers");
```

Which in this case would return myArray filtered and sorted by the EOFetchSpecification named "fetchUsers" which must be a model-based fetchspec in the first object's entity.

**FlattenOperator:** Define an NSArray.Operator for the key *flatten*.

This allows for key value paths like:

```
myArray.valueForKey("@flatten");
```

Which in this case would return myArray flattened if myArray is an NSArray of NSArray (of NSArray etc).

**IsEmptyOperator:** Define an NSArray.Operator for the key *isEmpty*.

This allows for key value paths like:

```
myArray.valueForKey("@isEmpty");
```

**SubarrayWithRangeOperator:** Define an NSArray.Operator for the key *subarrayWithRange*.

This allows for key value paths like:

```
myArray.valueForKey("@subarrayWithRange.3-20");
```

**UniqueOperator:** Define an NSArray.Operator for the key *unique*.

This allows for key value paths like:

```
myArray.valueForKeyPath("@unique.someOtherPath");
```

Which in this case would return only those objects which are unique in myArray.

**RemoveNullValuesOperator:** Define an NSArray.Operator for the key *removeNullValues*.

This allows for key value paths like:

```
myArray.valueForKeyPath("@removeNullValues.someOtherPath");
```

Which in this case would return myArray without the occurrences of NSKeyValueCoding.Null.

**ObjectAtIndexOperator:** Define an NSArray.Operator for the key *objectAtIndex*.

This allows for key value paths like:

```
myArray valueForKey:@"@objectAtIndex.3.firstName");
```

**AvgNonNullOperator:** Define an NSArray.Operator for the key *avgNonNull*.

This allows for key value paths like:

```
myArray valueForKey:@"@avgNonNull.revenue");
```

which will sum up all values and divide by the number of non-null entries.

**ReverseOperator:** Define an NSArray.Operator for the key *reverse*.

This allows for key value paths like:

```
myArray valueForKey:@"@reverse.someMorePath");
```

which return a reversed result as to you would normally get.

**MedianOperator:** Define an NSArray.Operator for the key *median*.

This allows for key value paths like:

```
myArray valueForKey:@"@median.someMorePath");
```

which return the median of the array elements at the given key path.

The median is the value for which half of the elements are above and half the elements are below. As such, an array sort is needed and this might be very costly depending of the size of the array.