Documentation

Language: **Swift**

Foundation / Filters and Sorting

API Collection

Filters and Sorting

Use predicates, expressions, and sort descriptors to examine elements in collections and other services.

Topics

Filltering

`struct Predicate`

A logical condition used to test a set of input values for searching or filtering.

`struct PredicateError`

An error thrown while evaluating a predicate.

`struct PredicateCodableConfiguration`

A specification of the expected types and key paths found in an archived predicate.

`protocol PredicateCodableKeyPathProviding`

A type that provides the expected key paths found in an archived predicate.

`protocol PredicateExpression`

A component expression that makes up part of a predicate.

`protocol StandardPredicateExpression`

A component expression that makes up part of a predicate, and that’s supported by the standard predicate type.

`enum PredicateExpressions`

The expressions that make up a predicate.

`struct PredicateBindings`

A mapping from a predicates's input variables to their values.

`class NSPredicate`

A definition of logical conditions for constraining a search for a fetch or for in-memory filtering.

`class NSExpression`

An expression for use in a comparison predicate.

`class NSComparisonPredicate`

A specialized predicate for comparing expressions.

`class NSCompoundPredicate`

A specialized predicate that evaluates logical combinations of other predicates.

Sorting

`class NSSortDescriptor`

An immutable description of how to order a collection of objects according to a property common to all the objects.

`enum ComparisonResult`

Constants that indicate sort order.

`struct SortDescriptor`

A serializable description of how to sort numerics and strings.

`protocol SortComparator`

A comparison algorithm for a specified type.

`struct ComparableComparator`

A comparator that compares types according to their conformance to the comparable protocol.

`struct KeyPathComparator`

A comparator that uses another sort comparator to provide the comparison of values at a key path.

`enum SortOrder`

The orderings that you can perform sorts with.

See Also

Fundamentals

- Numbers, Data, and Basic Values

  Work with primitive values and other fundamental types used throughout Cocoa.

- Strings and Text

  Create and process strings of Unicode characters, use regular expressions to find patterns, and perform natural language analysis of text.

- Collections

  Use arrays, dictionaries, sets, and specialized collections to store and iterate groups of objects or values.

- Dates and Times

  Compare dates and times, and perform calendar and time zone calculations.

- Units and Measurement

  Label numeric quantities with physical dimensions to allow locale-aware formatting and conversion between related units.

- Data Formatting

  Convert numbers, dates, measurements, and other values to and from locale-aware string representations.