Swift NSPredicateEditor Sample Code

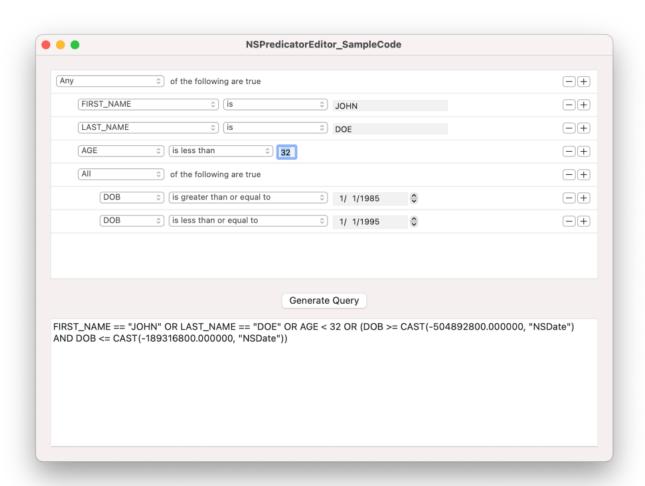
ddas.tech/swift-nspredicateeditor-sample-code/

November 19, 2022

Created By: Debasis Das (19-Nov-2022)

In this post we will create a simple Swift Application for Mac OS that demonstrates the use of NSPredicateEditor.

NSPredicateEditor is a powerful tool to form complex queries using a mix of compound predicates and comparison operators. This UI can be particularly used by developers to expose interaction with a database to an end user.



Swift – NSPredicateEditor sample code

```
// AppDelegate.swift
// NSPredicatorEditor_SampleCode
// Created by Debasis Das on 11/18/22.
import Cocoa
let defaultPredicate = "FIRST_NAME == 'JOHN' OR LAST_NAME == 'DOE' OR (DOB >=
CAST('1/1/1985 00:00', 'NSDate') AND DOB <= CAST('01/01/1995', 'NSDate'))"
@main
class AppDelegate: NSObject, NSApplicationDelegate {
    @IBOutlet var window: NSWindow!
    @IBOutlet weak var predicateEditor: NSPredicateEditor!
    @IBOutlet weak var queryTextField: NSTextField!
    func predicateRowTemplates()-> [NSPredicateEditorRowTemplate]{
        let compoundPredicateRowTemplate =
NSPredicateEditorRowTemplate(compoundTypes:
[NSNumber(value:NSCompoundPredicate.LogicalType.and.rawValue),
            NSNumber(value: NSCompoundPredicate.LogicalType.or.rawValue),
            NSNumber(value:NSCompoundPredicate.LogicalType.not.rawValue)]
        )
        let stringOperators = [
            NSNumber(value: NSComparisonPredicate.Operator.equalTo.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.notEqualTo.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.like.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.beginsWith.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.endsWith.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.contains.rawValue)
        1
        let dateAndNumberOperators = [
            NSNumber(value: NSComparisonPredicate.Operator.equalTo.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.notEqualTo.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.lessThan.rawValue),
            NSNumber(value:
NSComparisonPredicate.Operator.lessThanOrEqualTo.rawValue),
            NSNumber(value: NSComparisonPredicate.Operator.greaterThan.rawValue),
            NSNumber(value:
NSComparisonPredicate.Operator.greaterThanOrEqualTo.rawValue)
        1
        let firstNameRowTemplate = NSPredicateEditorRowTemplate(
            leftExpressions: [NSExpression(forKeyPath: "FIRST_NAME")],
            rightExpressionAttributeType: .stringAttributeType,
            modifier: .direct,
            operators: stringOperators,
            options: 0)
```

```
let lastNameRowTemplate = NSPredicateEditorRowTemplate(
            leftExpressions: [NSExpression(forKeyPath: "LAST_NAME")],
            rightExpressionAttributeType: .stringAttributeType,
            modifier: .direct,
            operators: stringOperators,
            options: 0)
        let dobRowTemplate = NSPredicateEditorRowTemplate(
            leftExpressions: [NSExpression(forKeyPath: "DOB")],
            rightExpressionAttributeType: .dateAttributeType,
            modifier: .direct,
            operators: dateAndNumberOperators,
            options: 0)
        let ageRowTemplate = NSPredicateEditorRowTemplate(
            leftExpressions: [NSExpression(forKeyPath: "AGE")],
            rightExpressionAttributeType: .integer32AttributeType,
            modifier: .direct,
            operators: dateAndNumberOperators,
            options: 0)
        return
[compoundPredicateRowTemplate, firstNameRowTemplate, lastNameRowTemplate, dobRowTemplate
, ageRowTemplate]
   }
    func applicationDidFinishLaunching(_ aNotification: Notification) {
        self.predicateEditor.rowTemplates.removeAll()
        self.predicateEditor.rowTemplates = self.predicateRowTemplates()
        self.predicateEditor.objectValue = NSPredicate(format: defaultPredicate)
   }
    @IBAction func generateQuery(_ sender:Any){
        self.queryTextField.stringValue = self.predicateEditor.predicate?.description
?? ""
    }
}
```

The available modifiers are direct, all, any

if direct modifier is used then a direct comparison is used.

Available Operators are as below

```
case lessThan = 0 // compare: returns NSOrderedAscending
case lessThanOrEqualTo = 1 // compare: returns NSOrderedAscending || NSOrderedSame
case greaterThan = 2 // compare: returns NSOrderedDescending
case greaterThanOrEqualTo = 3 // compare: returns NSOrderedDescending ||
NSOrderedSame
case equalTo = 4 // isEqual: returns true
case notEqualTo = 5 // isEqual: returns false
case matches = 6
case like = 7
case beginsWith = 8
case endsWith = 9
case `in` = 10 // rhs contains lhs returns true
case customSelector = 11
@available(macOS 10.5, *)
case contains = 99 // lhs contains rhs returns true
@available(macOS 10.5, *)
case between = 100
```

You can download the code from the below link.

https://github.com/debasisdas1/ddas.tech/tree/main/Swift/NSPredicatorEditor_SampleCode