INTERMEDIATE

CORE DATA

HANDS-ON CHALLENGES

Intermediate Core Data

Luke Parham

Notice of Rights

Notice of Liability

Trademarks

Challenge #1: Summing with NSExpression

By Luke Parham

In the demo you saw how to fetch the count of objects matching certain predicate.

Your challenge is to fill in the label for the **Offering a Deal** section in on the **Filters** screen.

Filling in the Method

To get started, navigate to **FilterViewController.swift** and scroll to the bottom where you'll see `populateDealsCountLabel()` waiting to be implemented.

The first thing to do is create a fetch request with `.dictionaryResultType` for its `resultType` property.

```
let fetchRequest = NSFetchRequest<NSDictionary>(entityName: "Venue")
fetchRequest.resultType = .dictionaryResultType
```

Normally, when you use a dictionary result type you specify an array of properties for which you want values returned.

In this case, you don't want the value of a property that currently exists on `Venue` types. Instead you're trying to sum together a number from each `Venue` to be combined and represented as a new property.

This is where `NSExpressionDescription` comes in.

```
let sumExpressionDesc = NSExpressionDescription()
sumExpressionDesc.name = "sumDeals"
sumExpressionDesc.expressionResultType = .integer32AttributeType
```

Here, you create an `NSExpressionDescription` and give it the name **sumDeals**. This is the "property" name you'll provide later. You also specifiy that you'd like the result as an `Int32`.

Next, add:

```
let specialCountExp = NSExpression(forKeyPath:
#keyPath(Venue.specialCount))
sumExpressionDesc.expression =
    NSExpression(forFunction: "sum:", arguments: [specialCountExp])
```

First, you create an `NSExpression` for the **specialCount** `keyPath`. Next you create another expression and tell it you want to use the `sum:` function and the input to the sum will be the special count of each Venue.

The last thing you need to do is add:

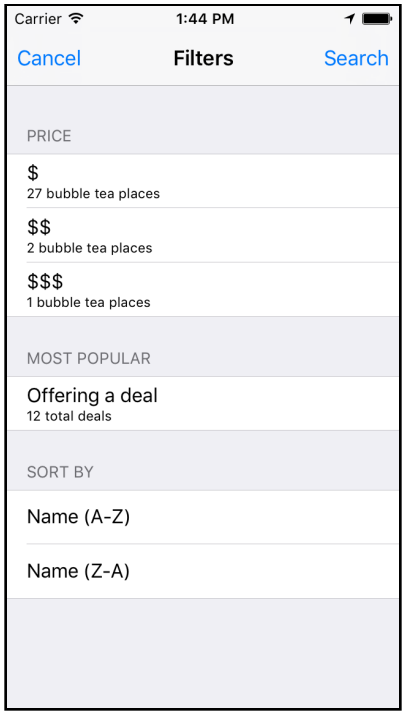
```
fetchRequest.propertiesToFetch = [sumExpressionDesc]

do {
  let results = try coreDataStack.managedContext.fetch(fetchRequest)
  let resultDict = results.first!
  let numDeals = resultDict["sumDeals"]!
  numDealsLabel.text = "\(numDeals) total deals"

} catch let error as NSError {
  print("Count not fetch \(error), \(error.userInfo)")
}
```

Here, you use `sumExpressionDesc` as the "property" to be used as the key in the resultant dicitonary.

Finally, you use the fetch request you built up and execute it. The result is an array with one value; a dictionary with the key `"specialCount"` pointing to the sum of all `specialCount` values.



To learn more about `NSExpression` check out the the api reference.