Lecture 11 Demo Code: Smashtag Core Data

Objective

Included below is the source code for the demo in lecture. It is provided under the same Creative Commons licensing as the rest of CS193p's course materials. Only the parts of `TweetTableViewController` that were changed (just making it more subclassable by adding the `insertTweets` method) is included (with surrounding unchanged code grayed out). See the Lecture 9 Smashtag Demo Code document for the rest of `TweetTableViewController`. The Twitter framework code is available via a link in the Assignment 4 writeup document. And here is the project (not including the Twitter framework or the workspace you'll need to contain them both).

```
//
//  TweetTableViewController.swift
//  Smashtag
//
//  Created by CS193p Instructor.
//  Copyright © 2017 Stanford University. All rights reserved.
//

class TweetTableViewController: UITableViewController, UITextFieldDelegate
{
   func insertTweets(_ newTweets: [Twitter.Tweet]) {
        self.tweets.insert(newTweets, at:0)
        self.tableView.insertSections([0], with: .fade)
    }

    // MARK: Updating the Table

   private func searchForTweets() {
       if let request = lastTwitterRequest?.newer ?? twitterRequest() {
            lastTwitterRequest = request
            request.fetchTweets { [weak self] newTweets in      // this is off the main queue
                DispatchQueue.main.async {                      // so dispatch back to main queue
                    if request == self?.lastTwitterRequest {
                        self?.insertTweets(newTweets)
                    }
                    self?.refreshControl?.endRefreshing()   // REFRESHING
                }
            }
        } else {
            self.refreshControl?.endRefreshing()            // REFRESHING
        }
    }
}
```

```
//
//  SmashTweetTableViewController.swift
//  Smashtag
//
//  Created by CS193p Instructor.
//  Copyright © 2017 Stanford University. All rights reserved.
//

import UIKit
import Twitter
import CoreData

class SmashTweetTableViewController: TweetTableViewController
{
    var container: NSPersistentContainer? =
        (UIApplication.shared.delegate as? AppDelegate)?.persistentContainer

    override func insertTweets(_ newTweets: [Twitter.Tweet]) {
        super.insertTweets(newTweets)
        updateDatabase(with: newTweets)
    }

    private func updateDatabase(with tweets: [Twitter.Tweet]) {
        print("starting database load")
        container?.performBackgroundTask { [weak self] context in
            for twitterInfo in tweets {
                _ = try? Tweet.findOrCreateTweet(matching: twitterInfo, in: context)
            }
            try? context.save()
            print("done loading database")
            self?.printDatabaseStatistics()
        }
    }

    private func printDatabaseStatistics() {
        if let context = container?.viewContext {
            context.perform {
                if Thread.isMainThread {
                    print("on main thread")
                } else {
                    print("off main thread")
                }
                // bad way to count
                if let tweetCount = (try? context.fetch(Tweet.fetchRequest()))?.count {
                    print("\(tweetCount) tweets")
                }
                // good way to count
                if let tweeterCount = try? context.count(for: TwitterUser.fetchRequest()) {
                    print("\(tweeterCount) Twitter users")
                }
            }
        }
    }

    // MARK: Navigation

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "Tweeters Mentioning Search Term" {
            if let tweetersTVC = segue.destination as? SmashTweetersTableViewController {
                tweetersTVC.mention = searchText
                tweetersTVC.container = container
            }
        }
    }
}
```

```swift
//
//  SmashTweetersTableViewController.swift
//  Smashtag
//
//  Created by CS193p Instructor.
//  Copyright © 2017 Stanford University. All rights reserved.
//

import UIKit
import CoreData

class SmashTweetersTableViewController: FetchedResultsTableViewController
{
    var mention: String? { didSet { updateUI() } }

    var container: NSPersistentContainer? =
        (UIApplication.shared.delegate as? AppDelegate)?.persistentContainer { didSet { updateUI() } }

    var fetchedResultsController: NSFetchedResultsController<TwitterUser>?

    private func updateUI() {
        if let context = container?.viewContext, mention != nil {
            let request: NSFetchRequest<TwitterUser> = TwitterUser.fetchRequest()
            request.sortDescriptors = [NSSortDescriptor(
                key: "handle",
                ascending: true,
                selector: #selector(NSString.localizedCaseInsensitiveCompare(_:))
            )]
            request.predicate = NSPredicate(format: "any tweets.text contains[c] %@", mention!)
            fetchedResultsController = NSFetchedResultsController<TwitterUser>(
                fetchRequest: request,
                managedObjectContext: context,
                sectionNameKeyPath: nil,
                cacheName: nil
            )
            fetchedResultsController?.delegate = self
            try? fetchedResultsController?.performFetch()
            tableView.reloadData()
        }
    }

    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "TwitterUser Cell", for: indexPath)

        if let twitterUser = fetchedResultsController?.object(at: indexPath) {
            cell.textLabel?.text = twitterUser.handle
            let tweetCount = tweetCountWithMentionBy(twitterUser)
            cell.detailTextLabel?.text = "\(tweetCount) tweet\((tweetCount == 1) ? "" : "s")"
        }

        return cell
    }

    private func tweetCountWithMentionBy(_ twitterUser: TwitterUser) -> Int {
        let request: NSFetchRequest<Tweet> = Tweet.fetchRequest()
        request.predicate = NSPredicate(format: "text contains[c] %@ and tweeter = %@", mention!, twitterUser)
        return (try? twitterUser.managedObjectContext!.count(for: request)) ?? 0
    }
}
```