```
//
//  AddressTableViewController.m
//  AddressBook
//
//  Created by yiner on 2018/6/20.
//  Copyright © 2018年 yiner. All rights reserved.
//

#import "AddressTableViewController.h"
#import "Address+Create.h"

@interface AddressTableViewController ()
@property (strong, nonatomic) IBOutlet UISearchController
*searchController;
@property (weak, nonatomic) IBOutlet UITableView *tableview;
// 搜索结果数组
@property (nonatomic, strong) NSMutableArray *results;
@end

@implementation AddressTableViewController
@synthesize searchController = _searchController;
@synthesize results = _results;

#pragma mark - Properties

- (void)setManagedObjectContext:(NSManagedObjectContext
*)managedObjectContext
{
    _managedObjectContext = managedObjectContext;
    if (managedObjectContext) {
        NSFetchRequest *request = [NSFetchRequest
fetchRequestWithEntityName:@"Address"];
        request.sortDescriptors = @[[NSSortDescriptor
sortDescriptorWithKey:@"name" ascending:YES
selector:@selector(localizedCaseInsensitiveCompare:)]];
        request.predicate = nil; // all Addresses
        self.fetchedResultsController =
[[NSFetchedResultsController alloc] initWithFetchRequest:request
managedObjectContext:managedObjectContext
sectionNameKeyPath:nil cacheName:nil];
    } else {
        self.fetchedResultsController = nil;
    }
```

```
}

- (NSMutableArray *)results {
      if (_results == nil) {
            _results = [NSMutableArray arrayWithCapacity:0];
      }

      return _results;
}

#pragma mark - View Controller Lifecycle

// Just sets the Refresh Control's target/action since it can't be set
in Xcode (bug?).

- (void)viewDidLoad
{
      [super viewDidLoad];
      [self.refreshControl addTarget:self
                                          action:@selector(refresh)

forControlEvents:UIControlEventValueChanged];

}

// Whenever the table is about to appear, if we have not yet opened/
created document, do so.

- (void)viewWillAppear:(BOOL)animated
{
      [super viewWillAppear:animated];
      if (!self.managedObjectContext) [self useDocument];

      if (self.searchController == nil) {
            // 加载搜索框
            UISearchController *search = [[UISearchController
alloc] initWithSearchResultsController:nil];
            // 设置结果更新代理
            search.searchResultsUpdater = self;
            // 因为在当前控制器展示结果, 所以不需要这个透明视图
            search.dimsBackgroundDuringPresentation = NO;
```

```
        // 当搜索框激活时，不隐藏导航条
        search.hidesNavigationBarDuringPresentation = NO;
        self.searchController = search;
        // 将searchBar赋值给tableView的tableHeaderView
        self.tableView.tableHeaderView = search.searchBar;
    }
    self.searchController.searchBar.hidden = NO;

}

- (void)useDocument
{
    NSURL *url = [[[NSFileManager defaultManager]
URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask]
lastObject];
    url = [url URLByAppendingPathComponent:@"Address Document"];
    UIManagedDocument *document = [[UIManagedDocument alloc]
initWithFileURL:url];

    if (![[NSFileManager defaultManager] fileExistsAtPath:[url
path]]) {
        [document saveToURL:url
           forSaveOperation:UIDocumentSaveForCreating
          completionHandler:^(BOOL success) {
              if (success) {
                  self.managedObjectContext =
document.managedObjectContext;
                  [self refresh];
              }
          }];
    } else if (document.documentState ==
UIDocumentStateClosed) {
        [document openWithCompletionHandler:^(BOOL
success) {
            if (success) {
                self.managedObjectContext =
document.managedObjectContext;
                [self refresh];
            }
        }];
    } else {
        self.managedObjectContext =
document.managedObjectContext;
        [self refresh];
    }
}

#pragma mark - Refreshing

- (IBAction)refresh
{
    [self.refreshControl beginRefreshing];
    dispatch_queue_t fetchQ = dispatch_queue_create("Address
Fetch", NULL);
    dispatch_async(fetchQ, ^{
        NSString *path = [[NSBundle mainBundle]
pathForResource:@"address" ofType:@"txt"];
        NSString *contentStr = [NSString
stringWithContentsOfFile:path encoding:NSUTF8StringEncoding
error:nil];
        NSArray *info = [contentStr
componentsSeparatedByString:@"\n"];

        // put the addresses in Core Data
        [self.managedObjectContext performBlock:^{
            NSDictionary *address = nil;
            for (NSString *line in info){
                if (![line isEqual:@""]){
                    NSArray *items = [line
componentsSeparatedByString:@"\t"];
                    address = @{
                        @"name" : items[0],
                        @"phone" : items[1],
                        @"email" : items[2],
                        @"address" : items[3],
                        @"weChat" : items[4],
                        //@"image" : items[5]
                        };
                    [Address
insertDataWithAddressInfo:address
inManagedObjectContext:self.managedObjectContext];
                }
            }

            dispatch_async(dispatch_get_main_queue(), ^{
                [self.refreshControl endRefreshing];
            });
        }];
    });
}

#pragma mark - UITableViewDataSource

- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    // 这里通过searchController的active属性来区分展示数据源是哪个
    if (self.searchController.active) return self.results.count ;

    return [[[self.fetchedResultsController sections]
objectAtIndex:section] numberOfObjects];
}

// 设置单元格

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath {

    UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:@"Address Cell"];
    if (!cell) {
        cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault

reuseIdentifier:@"Address Cell"];
    }
    Address *address = nil;
    // 这里通过searchController的active属性来区分展示数据源是哪个
    if (self.searchController.active ) {

        address = [self.results
objectAtIndex:indexPath.row];
    } else {

        address = [self.fetchedResultsController
objectAtIndexPath:indexPath];
    }

    //Address *address = [self.fetchedResultsController
objectAtIndexPath:indexPath];

    cell.textLabel.text = address.name;
    NSLog(@"%@",address.name);
    //cell.imageView = address.image;
    return cell;
}

#pragma mark - UISearchResultsUpdating

- (void)updateSearchResultsForSearchController:
(UISearchController *)searchController {

    NSString *inputStr = searchController.searchBar.text
;
    if (self.results.count > 0) {
        [self.results removeAllObjects];
    }
    for (Address *address in [self.fetchedResultsController
fetchedObjects]) {

        if ([address.name.lowercaseString
rangeOfString:inputStr.lowercaseString].location != NSNotFound)
{
```

```
            [self.results addObject:address];
        }
    }

    [self.tableView reloadData];
}

#pragma mark - Segue

- (void)prepareForSegue:(UIStoryboardSegue *)segue
sender:(id)sender
{
    NSIndexPath *indexPath = nil;

    // segue to EditVC
    if ([segue.identifier isEqualToString:@"add address"]) {
        if ([segue.destinationViewController
respondsToSelector:@selector(setSourceType:)]) {
            [segue.destinationViewController
performSelector:@selector(setSourceType:) withObject:@"add"];
        }
        // set context
        if ([segue.destinationViewController
respondsToSelector:@selector(setManagedObjectContext:)]) {
            [segue.destinationViewController
performSelector:@selector(setManagedObjectContext:)
withObject:self.managedObjectContext];
        }
    }

    //segue to DetailVC
    if ([sender isKindOfClass:[UITableViewCell class]]) {
        indexPath = [self.tableView
indexPathForCell:sender];
    }

    if (indexPath) {
        if ([segue.identifier isEqualToString:@"show address
detail"]) {
```

```
            Address *address = nil;
            if ([self.results count] > 0) {
                address = [self.results
objectAtIndex:indexPath.row];
                self.results = nil;
            }
            else address = [self.fetchedResultsController
objectAtIndexPath:indexPath];
            // 如果目标VC有这个方法(selector)就让它执行这个方法
            if ([segue.destinationViewController
respondsToSelector:@selector(setAddress:)]) {
                [segue.destinationViewController
performSelector:@selector(setAddress:) withObject:address];
            }
            // set context
            if ([segue.destinationViewController
respondsToSelector:@selector(setManagedObjectContext:)]) {
                [segue.destinationViewController
performSelector:@selector(setManagedObjectContext:)
withObject:self.managedObjectContext];
            }
        }
    }

    // 取消搜索栏
    self.searchController.active = NO;

}

@end
```