```swift
//
//  EventTalksController.swift
//  Eventor
//
//  Created by Rostyslav Kobizsky on 11/21/14.
//  Copyright (c) 2014 Rozdoum. All rights reserved.
//

import UIKit


class EventTalksController: UIViewController, CBLGUITableDelegate,
    UITableViewDelegate, UISearchBarDelegate, SWTableViewCellDelegate,
    UISplitViewControllerDelegate, EventController {

    var event: Event!

    private var appDelegate : AppDelegate {
        return UIApplication.sharedApplication().delegate as AppDelegate
    }
    private var scheduledLiveQuery: CBLLiveQuery!

    @IBOutlet private var dataSource: CBLGUITableSource!
    @IBOutlet private var tableView: UITableView!
    @IBOutlet weak var searchBar: UISearchBar!

    override func viewDidLoad() {
        super.viewDidLoad()

        assert(event != nil, "Event can't be nil")
        navigationItem.leftBarButtonItem = ThemeFactory.defaultTheme.
            menuBarButtonItem(self, action: "openMenu:")

        dataSource.grouped = "dateFormattedString"
        dataSource.sortDescriptors = [NSSortDescriptor(key: "start",
            ascending: false)]
        dataSource.liveQuery = Talk.query(event: event).asLiveQuery()
        dataSource.liveQuery?.start()
        searchDisplayController?.searchResultsTableView.dataSource =
            dataSource
    }

    override func viewWillAppear(animated: Bool) {
        super.viewWillAppear(animated)
        if let indexPaths = tableView.indexPathsForSelectedRows() as?
            [NSIndexPath] {
            for indexPath in indexPaths {
                tableView.deselectRowAtIndexPath(indexPath, animated:
                    animated)
            }
        }
    }

    func positionForBar(bar: UIBarPositioning) -> UIBarPosition {
        return .Top
    }

    // MARK: CBLGUITableDelegate

    func couchTableSource(source: CBLGUITableSource, cellForRowAtIndexPath
```

```
    indexPath: NSIndexPath) -> UITableViewCell {
    let cell =
        source.tableView?.dequeueReusableCellWithIdentifier("TalkCellIdenti
        fier", forIndexPath: indexPath) as TalkTableViewCell

    if let talk = source.objectAtIndexPath(indexPath) as? Talk {
        cell.talk = talk
        cell.delegate = self
    }

    return cell
}

func couchTableSource(source: CBLGUITableSource, titleForHeaderInSection
    section: Int) -> String? {
    if let talk = source.objectAtIndexPath(NSIndexPath(forRow: 0,
        inSection: section)) as? Talk {
        return talk.dateFormattedString()
    }
    return nil
}

func couchTableSourceDidChangeModel(source: CBLGUITableSource) {

    if (searchDisplayController?.active == true) {
        searchDisplayController?.searchResultsTableView.reloadData()
    } else {
        tableView.reloadData()
    }
}

func tableView(tableView: UITableView, estimatedHeightForRowAtIndexPath
    indexPath: NSIndexPath) -> CGFloat {
    return UITableViewAutomaticDimension
}

func tableView(tableView: UITableView, heightForRowAtIndexPath indexPath:
    NSIndexPath) -> CGFloat {
    return UITableViewAutomaticDimension
}

// MARK: UISearchBar Delegate

func searchBarTextDidBeginEditing(searchBar: UISearchBar) {
    searchBar.setShowsCancelButton(true, animated: true)
}

func searchBarTextDidEndEditing(searchBar: UISearchBar) {
    searchBar.setShowsCancelButton(false, animated: true)
}

func searchBarCancelButtonClicked(searchBar: UISearchBar) {
    searchBar.resignFirstResponder()
    self.dataSource.predicate = nil
}

func searchBar(searchBar: UISearchBar, textDidChange searchText: String) {
    if searchText != "" {
        self.dataSource.predicate = NSPredicate(format: "title CONTAINS[c]
            %@", searchText)
```

```
    }
    else {
        self.dataSource.predicate = nil
    }
}

func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath:
    NSIndexPath) {
    tableView.deselectRowAtIndexPath(indexPath, animated: true)
    self.performSegueWithIdentifier("ShowTalk", sender: tableView.
        cellForRowAtIndexPath(indexPath))
}

// MARK: Actions

@IBAction func dismissSplitController(sender: AnyObject?) {
    splitViewController?.evo_drawerController?.openDrawerSide(.Left,
        animated: true, completion: nil)
}

// MARK: - SWTableViewCell delegate

func swipeableTableViewCellShouldHideUtilityButtonsOnSwipe(cell:
    SWTableViewCell!) -> Bool {
    return true
}

func swipeableTableViewCell(cell: SWTableViewCell!,
    didTriggerRightUtilityButtonWithIndex index: Int) {
    let appDelegate = UIApplication.sharedApplication().delegate as
        AppDelegate
    let talk = (cell as TalkTableViewCell).talk

    let detailsNavController =
        self.splitViewController?.viewControllers[1] as?
        UINavigationController

    switch (index) {
    case 0: // Share
        if detailsNavController != nil {
            let storyboard = UIStoryboard(name: "Main", bundle: nil)
            let shareController = storyboard.
                instantiateViewControllerWithIdentifier("shareController")
                as EventTalkShareController
            shareController.talk = talk
            detailsNavController!.pushViewController(shareController,
                animated: true)
        }
        else {
            self.performSegueWithIdentifier("showShareController", sender:
                talk)
        }
    case 1: // Feedback
        if detailsNavController != nil {
            let storyboard = UIStoryboard(name: "Main", bundle: nil)
            let shareController = storyboard.
                instantiateViewControllerWithIdentifier("feedbackController
                ") as EventTalkFeedbackController
            shareController.talk = talk
            detailsNavController!.pushViewController(shareController,
```

```
                    animated: true)
            }
            else {
                self.performSegueWithIdentifier("showFeedback", sender: talk)
            }
        case 2: // Schedule
            let agendaDocID = "agenda:" + appDelegate.profile.user_id + ":" +
                talk.document.documentID
            let agenda = AgendaTalk(talk: talk!, owner: appDelegate.profile)
            agenda.scheduled = !(cell.rightUtilityButtons[index] as UIButton).
                selected

            var error: NSError?
            assert(agenda.save(&error), "Can't save model into database!")
        default:
            return
        }

        cell.hideUtilityButtonsAnimated(true)
    }

    // MARK: Navigation

    override func prepareForSegue(segue: UIStoryboardSegue, sender:
        AnyObject?) {

        switch segue.destinationViewController {
        case let navigationController as UINavigationController:
            if let controller = navigationController.viewControllers.first as?
                EventTalkController {
                if let cell = sender as? TalkTableViewCell {
                    controller.talk = cell.talk
                }
            }
        case is EventTalkFeedbackController:
            if sender != nil {
                (segue.destinationViewController as
                    EventTalkFeedbackController).talk = sender as Talk
            }
        case is EventTalkShareController:
            if sender != nil {
                (segue.destinationViewController as EventTalkShareController).
                    talk = sender as Talk
            }
        default:
            return
        }
    }
}
```