

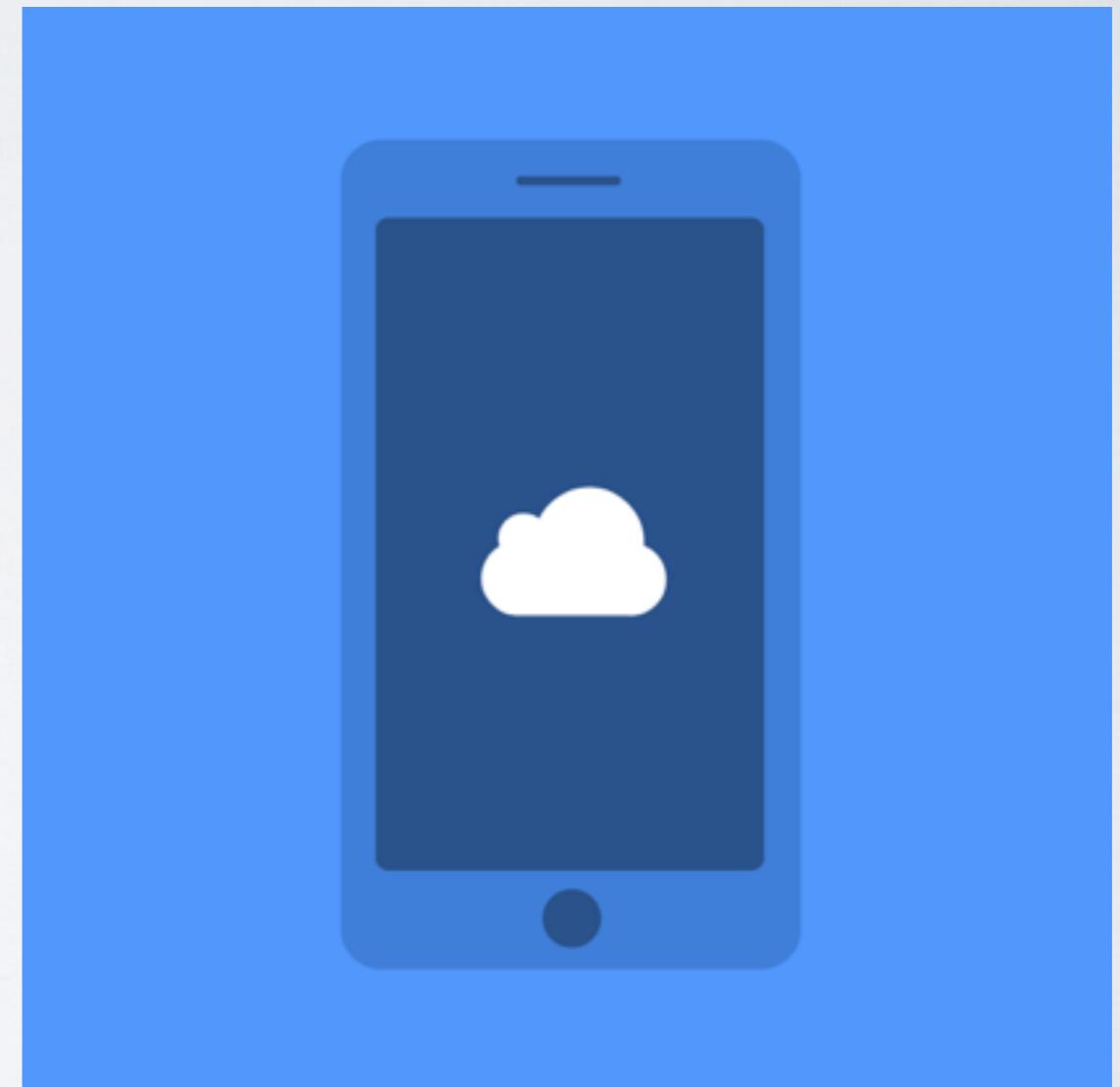


TABLEVIEW WITH PARSE SDK

Nick Chen
Fall 2015
talentspark.io

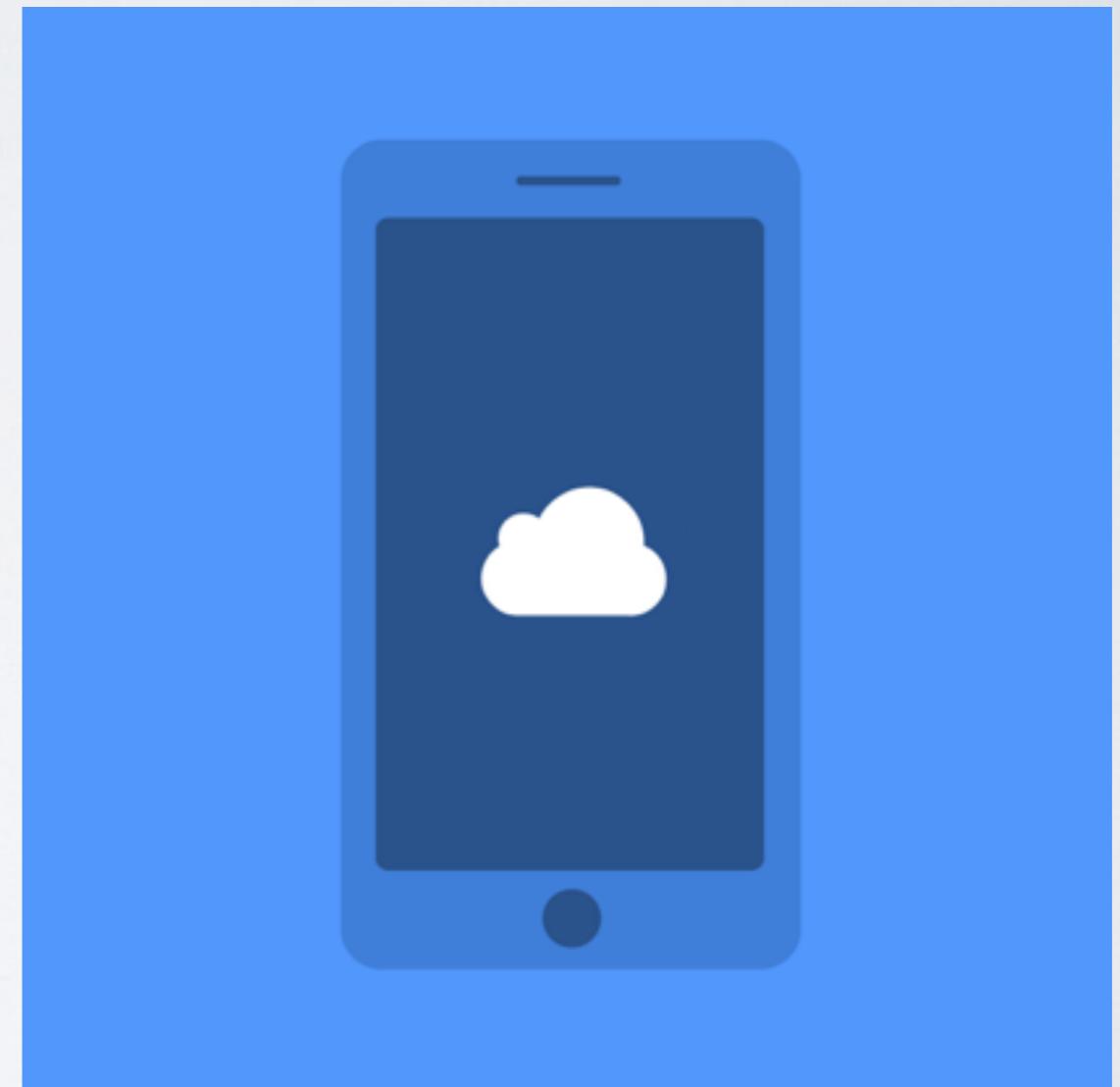
PARSE SDK

- Cross-platform SDK that provides cloud-based CRUD capabilities for your app
- Simplifies lots of distributed app code
- Generous free tier
- Good to consider for your final project



PARSE SDK

- Cross-platform SDK that provides cloud-based CRUD capabilities for your app
- Simplifies lots of distributed app code
- Generous free tier
- Good to consider for your final project



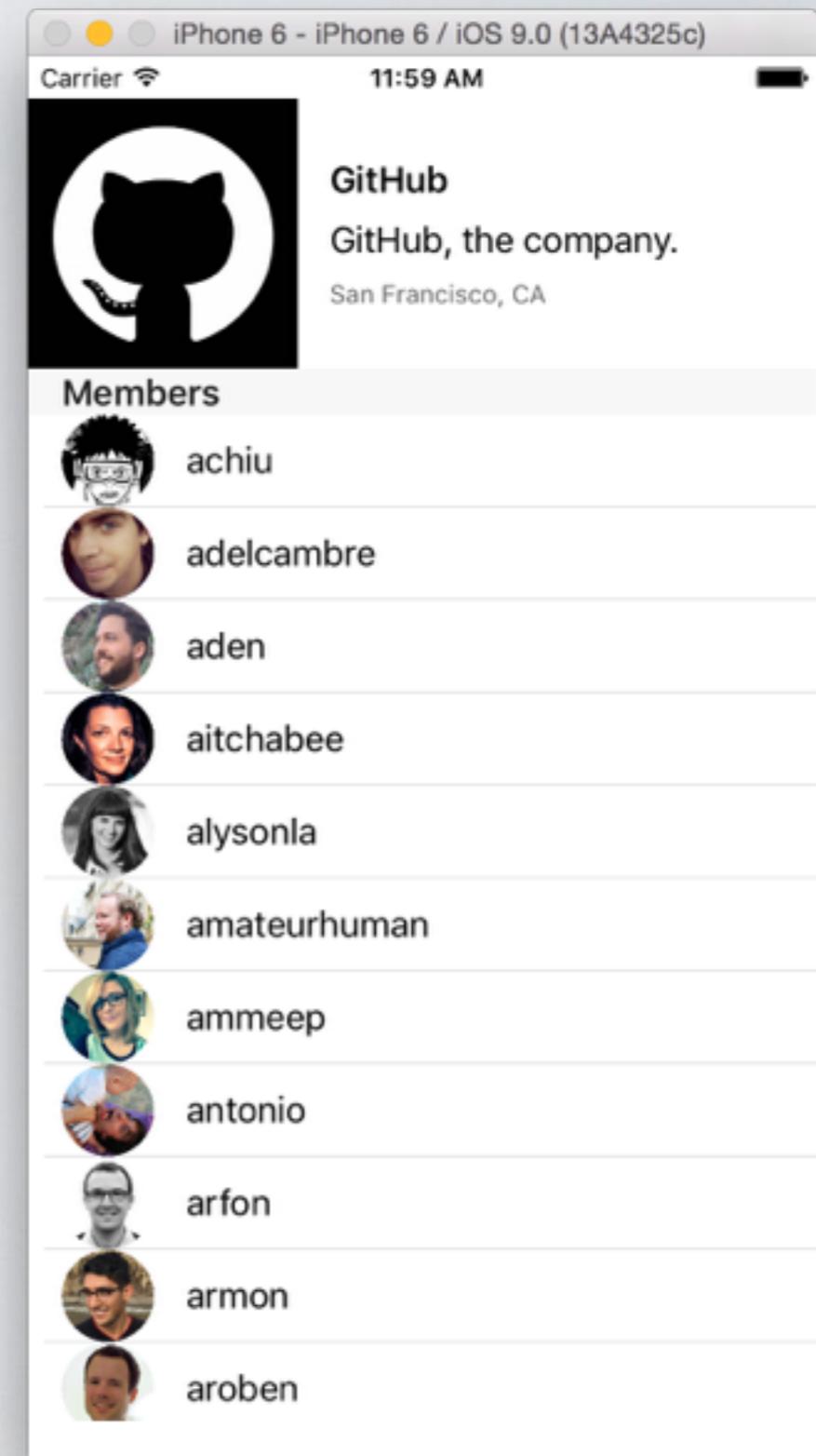
CLOUDKIT

- Apple SDK that provides cloud-based CRUD capabilities for your app
- Simplifies lots of distributed app code
- Very generous free tier
- Also good to consider for your final project



PREVIOUSLY

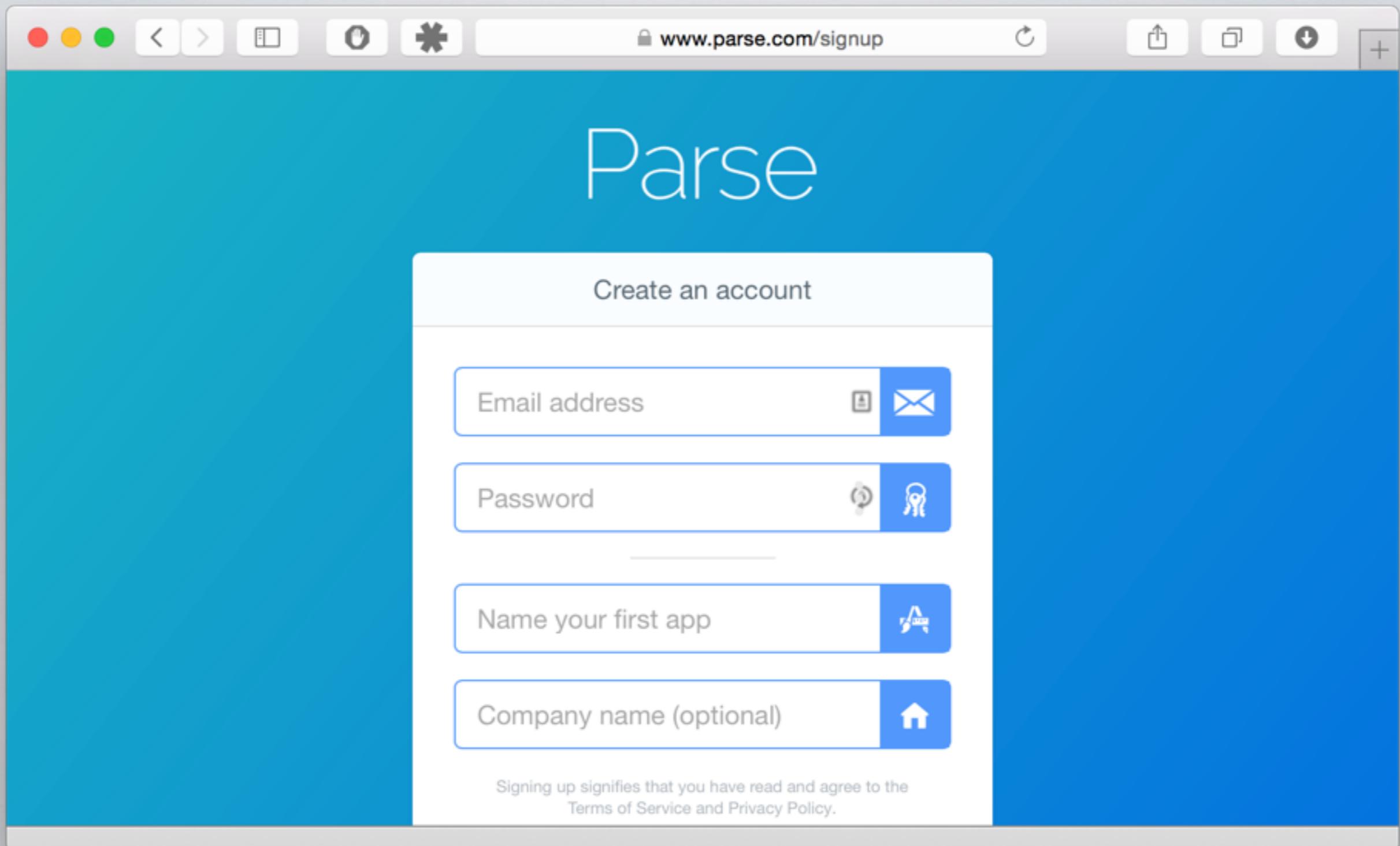
- We **Read** (part of **CRUD**) GitHub data
- Let's add **Create**, **Update** and **Delete** capabilities
- Could do this with GitHub API but let's use our own objects



CLOUDNOTE

- Yeah, yet-another-note-taking app
- But you will learn about Parse SDK and finish up what you learned before about TableView

PARSE



PARSE

The screenshot shows the Parse Dashboard interface. At the top, there's a navigation bar with standard OS X-style icons (red, yellow, green buttons, back/forward, search, etc.), a URL field showing www.parse.com/apps, and a toolbar with download/upload/copy/paste/refresh buttons. On the left, a sidebar has a blue header with a 'P' icon and the text 'Select an App'. Below it is a dropdown menu. To the right of the sidebar is the main 'Dashboard' area with a large 'Create a new App' button. Two app cards are displayed: 'CloudNote' (DEV status, created about 1 hour ago) and 'Paws' (DEV status, created 17 days ago). Both cards feature a map icon, a gear icon for settings, and a 'Just getting started? Check out the [quickstart guide!](#)' message. At the bottom, a dark footer bar contains links for 'Docs', 'Billing', 'Downloads', 'Help', and 'Status'.

Select an App

Dashboard

Create a new App

DEV

CloudNote

Created about 1 hour ago

Just getting started?
Check out the [quickstart guide!](#)

DEV

Paws

Created 17 days ago

Just getting started?
Check out the [quickstart guide!](#)

Docs Billing Downloads Help Status

PARSE

The screenshot shows the Parse Dashboard interface. At the top, there's a navigation bar with standard OS X-style icons (red, yellow, green circles, back, forward, etc.) and a URL bar showing www.parse.com/apps. Below the bar, a blue header bar says "Select an App" with a dropdown arrow and "Dashboard" on the right, along with a user profile icon.

In the center, there are two app cards:

- CloudNote**: Status: DEV. Created about 1 hour ago. Includes a "Just getting started? Check out the [quickstart guide!](#)" message.
- Paws**: Status: DEV. Created 17 days ago. Includes a "Just getting started? Check out the [quickstart guide!](#)" message.

A prominent green callout box with a white border and a triangular pointer on the left side points to the "Create a new App" button, which has a blue plus sign icon and the text "Create a new App".

At the bottom of the dashboard, there's a dark footer bar with links: Docs, Billing, Downloads, Help, and Status.

PARSE

A screenshot of the Parse.com web interface, specifically the Data tab for the CloudNote app in DEV mode. The interface includes a top navigation bar with standard OS X-style buttons and a URL bar showing the current page. Below the navigation is a dark header bar with the app name "CloudNote", a "Core" button, and several other icons. The main content area has a blue header "Data" with buttons for "+ Row", "- Row", "+ Col", "Security", and "More". A large central icon represents a grid. The text "No classes to display" is prominently displayed, followed by the sub-instruction "In order to store data in Parse, you'll need to make a class." On the left, a sidebar lists other tabs: "Cloud Code", "Webhooks", "Jobs", "Logs", "Config", and "API Console". At the bottom right, there are pagination controls for "20" rows. The footer contains links for "Docs", "Billing", "Downloads", "Help", and "Stats".

www.parse.com/apps/cloudnote--4/collections

P CloudNote DEV Core

Data + Row - Row + Col Security More

+ Add Class Import

Cloud Code

Webhooks

Jobs

Logs

Config

API Console

No classes to display

In order to store data in Parse, you'll need to make a class.

20 rows

Docs Billing Downloads Help Stats

Open # on this page in a new tab

PARSE

A screenshot of the Parse.com web interface, specifically the Data tab for the 'CloudNote' application in 'DEV' mode. The interface includes a top navigation bar with standard browser controls, a Parse-specific header with icons for Core, Security, and Settings, and a user profile picture. The left sidebar contains links for Data, Cloud Code, Webhooks, Jobs, Logs, Config, and API Console. The main content area shows a large green button labeled 'Add a new class' with a green callout pointing to it. Below this button is a large icon of a grid. The text 'No classes to display' is centered, followed by the sub-instruction 'In order to store data in Parse, you'll need to make a class.' At the bottom right, there are pagination controls for '20' rows and links for Docs, Billing, Downloads, Help, and Status.

www.parse.com/apps/cloudnote--4/collections

P CloudNote DEV Core

Data + Add Class Import

Cloud Code

Webhooks

Jobs

Logs

Config

API Console

Add a new class

No classes to display

In order to store data in Parse, you'll need to make a class.

20 rows

Docs Billing Downloads Help Status

Open # on this page in a new tab

PARSE

The screenshot shows the Parse.com interface with the following details:

- Header:** www.parse.com/apps/cloudnote--4/collections
- Left Sidebar (Data Tab):**
 - + Add Class
 - Import
- Center Content:** A modal window titled "Add a new class" is open. It contains:
 - A dropdown menu set to "Custom".
 - An input field containing the class name "Note".
 - A note: "Class names must only contain numbers, letters, and underscore, and can only begin with a letter."
 - Buttons: "Create Class" (blue) and "Cancel".
- Main Area:** The text "No classes to display" is centered, with a sub-note below it: "In order to store data in Parse, you'll need to make a class."
- Bottom Right:** Pagination controls (20, 50, 100) and links for Docs, Billing, Downloads, Help, and Status.

PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a toolbar with standard OS X window controls (red, yellow, green buttons), a refresh icon, a search bar containing the URL "www.parse.com/apps/cloudnote--4/collections#", and several other icons for file operations.

The main header includes the project name "CloudNote" and a "DEV" dropdown. To the right of the header are icons for "Core" (with a gear icon), "Logs" (with a waveform icon), "Comments" (with a speech bubble icon), "Settings" (with a gear icon), and "Books" (with a book icon).

The left sidebar has a "Data" tab selected, showing a table structure for the "Note" class. The table has columns: "objectId" (String), "Note" (0 rows), and "updatedAt" (Date). There are buttons for "+ Row", "- Row", "+ Col", "Security", "More", and a filter icon. A prominent button "Add a new column" is highlighted with a blue border.

Below the table, there are buttons for "+ Add Class" and "(+) Import".

The sidebar also lists other sections: "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config".

The central area displays the message "No data to display" with a sub-instruction "Add a row to store an object in this class." Below this, there's a pagination control showing "20 rows/page" and a back/forward navigation icon.

At the bottom, there are links for "Docs", "Billing", "Downloads", "Help", and "Status".

PARSE

The screenshot shows the Parse.com web interface for a project named "CloudNote". The main navigation bar includes icons for Core, Security, and Settings. Below the navigation is a toolbar with buttons for "+ Row", "- Row", "+ Col", "Security", "More", and a filter icon.

The left sidebar lists various project components: Data, Cloud Code, Webhooks, Jobs, Logs, and Config. The "Data" section shows a table for the "Note" class, which has 0 rows. The table includes columns for objectId, createdAt, and updatedAt.

A modal dialog titled "Add a Column" is open in the center. It allows selecting the column type (String) and entering a title ("Title"). A note below the title field specifies that the name must begin with a letter or number and contain alphanumeric or underscore characters. There are "Create Column" and "Cancel" buttons at the bottom of the dialog.

At the bottom of the screen, there are links for Docs, Billing, Downloads, Help, and Status, along with a row selection dropdown set to 20 rows per page.

PARSE

The screenshot shows the Parse.com web interface for a project named "CloudNote". The "Data" tab is selected, displaying a table with columns: Note (0), objectId (String), Title (String), createdAt (Date), updatedAt (Date). A modal window titled "Add a Column" is open, showing a "String" dropdown and a text input field containing "Body". Below the input field is a validation message: "Must only contain alphanumeric or underscore characters, and must begin with a letter or number." At the bottom of the modal are "Create Column" and "Cancel" buttons.

www.parse.com/apps/cloudnote--4/collections#

P CloudNote DEV Core

Data + Row - Row + Col Security More

Note 0 objectId String Title String createdAt Date updatedAt Date

+ Add Class Import

Cloud Code Webhooks Jobs Logs Config

Add a Column

String Body

Must only contain alphanumeric or underscore characters, and must begin with a letter or number.

Create Column Cancel

display in this class.

20 rows/page

Docs Billing Downloads Help Status

PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a navigation bar with standard OS X-style buttons (red, yellow, green) and a URL bar showing www.parse.com/apps/cloudnote--4/collections#. Below the navigation bar is a header with the project name "CloudNote" and a "DEV" dropdown. To the right of the header are icons for Core, Analytics, Chat, Settings, and Help.

The main area is titled "Data" and shows a table for the "Note" class. The columns listed are "objectId" (String), "Title" (String), "createdAt" (Date), and "updatedAt" (Date). A modal window titled "Add a Column" is open in the center. Inside the modal, there's a dropdown menu set to "File" and a text input field containing "Photo". Below these fields is a note: "Must only contain alphanumeric or underscore characters, and must begin with a letter or number." At the bottom of the modal are two buttons: "Create Column" (in blue) and "Cancel".

On the left side of the dashboard, there's a sidebar with links: "+ Add Class", "(+) Import", "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config". At the bottom right of the main area, there are buttons for "20 rows/page" and navigation arrows. The bottom navigation bar includes links for "Docs", "Billing", "Downloads", "Help", and "Stats".

PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a toolbar with standard OS X-style buttons (red, yellow, green) and a tab bar showing 'CloudNote' and 'DEV'. Below the toolbar, there's a navigation menu with 'Data' selected, followed by buttons for '+ Row', '- Row', '+ Col', 'Security', 'More', and a filter icon.

The main area displays a table for the 'Note' class. The columns are listed as 'objectId String', 'Title String', 'createdAt Date', and 'updatedAt Date'. A modal window titled 'Add a Column' is open, showing a dropdown menu set to 'File' with an option 'Photo' highlighted. A green callout bubble with the text 'For binary data' points to the 'File' dropdown. Below the dropdown, there's a note: 'Must only contain alphanumeric or underscore characters, and must begin with a letter or number.' To the right of the modal, the word 'display' is partially visible, along with 'in this class.' At the bottom of the modal, there are 'Column' and 'Cancel' buttons. The bottom of the screen shows a footer with links for 'Docs', 'Billing', 'Downloads', 'Help', and 'Stats', and a row for selecting 'rows/page' (set to 20).

PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a toolbar with standard OS X window controls (red, yellow, green buttons), a refresh icon, a search bar containing the URL "www.parse.com/apps/cloudnote--4/collections#/", and several other icons for file operations. Below the toolbar is a header with the project name "CloudNote" and a "DEV" dropdown. To the right of the header are icons for "Core", "Analytics", "Cloud Code", "Jobs", "Logs", and "Config". A user profile picture is also present.

The main area is divided into sections. On the left, a sidebar has a "Data" tab selected, which is highlighted in blue. Other tabs include "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config". Below the tabs are buttons for "+ Add Class" and "(+) Import".

The central part of the screen displays a table for the "Note" class. The table has columns: Note (with a count of 1), objectId (String), Title (String), createdAt (Date), and updatedAt (Date). There are two rows of data:

Note	objectId	Title	createdAt	updatedAt
1	ug9aWR7lUd	My First Note!	Aug 09, 2015, 04:12	Aug 09, 2015, 04:13

Below the table are buttons for "+ Row" (add a new row), "- Row" (remove a row), "+ Col" (add a new column), "Security" (set security rules), "More" (dropdown menu), and a filter icon. At the bottom of the main area, there are buttons for "20" rows/page and a back/forward navigation icon. The footer contains links for "Docs", "Billing", "Downloads", "Help", and "Status".

PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a navigation bar with standard OS X window controls (red, yellow, green buttons), a title bar with the URL www.parse.com/apps/cloudnote--4/collections#, and a toolbar with icons for Core, Analytics, Chat, Settings, and Help.

The main area displays a table titled "CloudNote" with one row visible. The columns are labeled "Note", "objectId", "content", "Title", "createdAt", and "updatedAt". The first row has values: Note "1", objectId "55a3e03f1f3d4a0001000001", content "CloudNote String", Title "String", createdAt "Aug 09, 2015, 04:12", and updatedAt "Aug 09, 2015, 04:13".

On the left sidebar, under the "Data" tab, there are buttons for "+ Add Class" and "(+) Import". Below the sidebar, there are links for "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config".

A green callout box with the text "Try adding a row" is overlaid on the "+ Row" button in the toolbar.

At the bottom right, there are buttons for "20 rows/page" and navigation arrows. The footer includes links for "Docs", "Billing", "Downloads", "Help", and "Status".

Note	objectId	content	Title	createdAt	updatedAt
1	55a3e03f1f3d4a0001000001	CloudNote String	String	Aug 09, 2015, 04:12	Aug 09, 2015, 04:13

PARSE

Note	Type
Title	String
Body	String
Photo	File
ObjectId	Id
CreatedAt	Date
UpdatedAt	Date
ACL	Permissions

PARSE

Note	Type
Title	String
Body	String
Photo	File
ObjectId	Id
CreatedAt	Date
UpdatedAt	Date
ACL	Permissions

Our own properties

PARSE

Note	Type
Title	String
Body	String
Photo	File
ObjectId	Id
CreatedAt	Date
UpdatedAt	Date
ACL	Permissions

Our own properties

Parse provided
properties

SKELETON CODE

```
● ● ● 2. vazexqi@daigo: ~/Development/CloudNote (zsh)
vazexqi@daigo:~/Development|⇒ git clone git@github.com:talentsparkio/CloudNote.git 21:56:10
Cloning into 'CloudNote'...
remote: Counting objects: 242, done.
remote: Compressing objects: 100% (154/154), done.
remote: Total 242 (delta 44), reused 242 (delta 44), pack-reused 0
Receiving objects: 100% (242/242), 9.43 MiB | 2.10 MiB/s, done.
Resolving deltas: 100% (44/44), done.
Checking connectivity... done.
vazexqi@daigo:~/Development|⇒ cd CloudNote
vazexqi@daigo:~/Development/CloudNote|⇒ git reset --hard InitialImport 21:56:19
± master ✓ 21:56:19
```

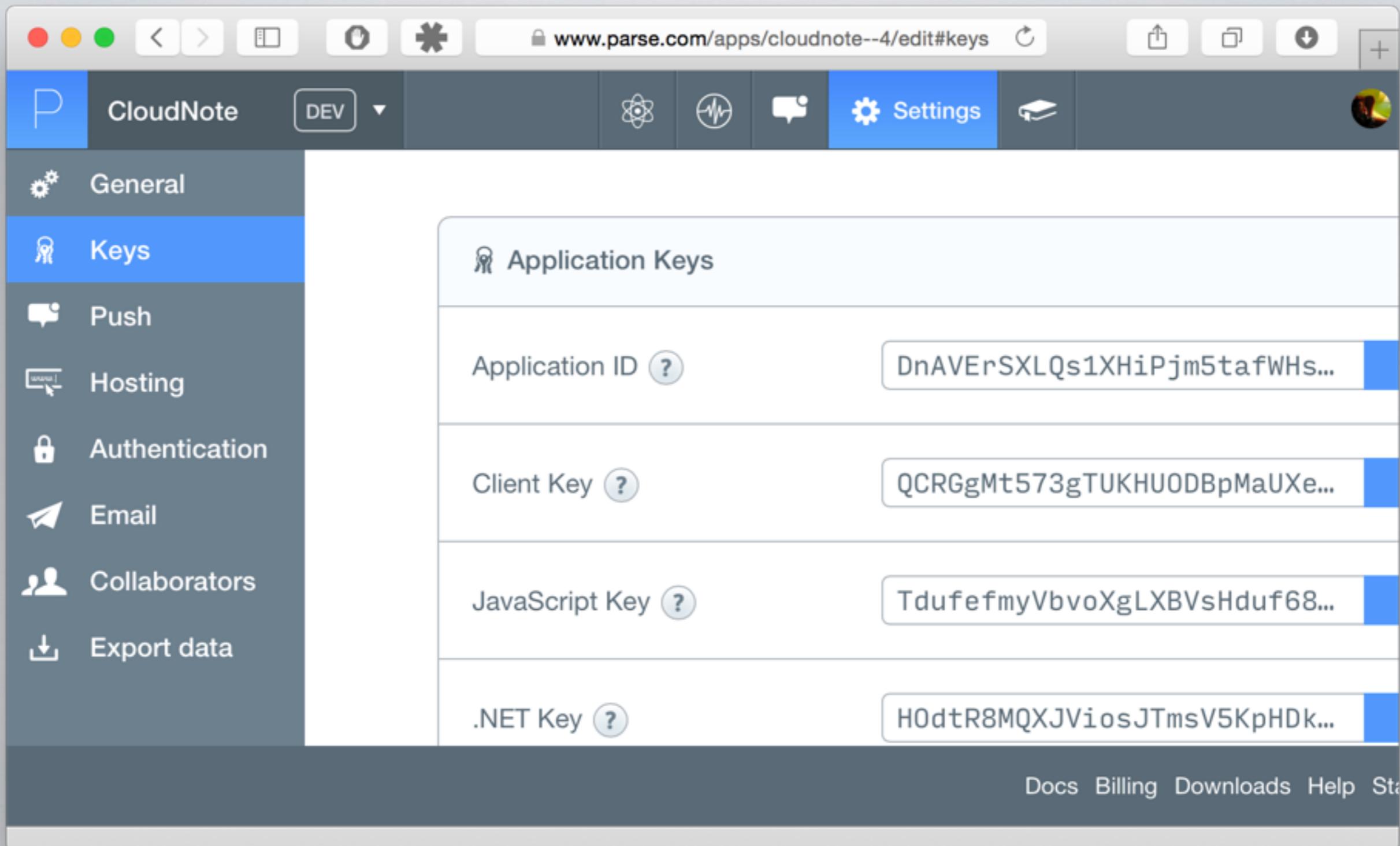
SKELETON CODE

```
2. vazexqi@daigo: ~/Development/CloudNote (zsh)
vazexqi@daigo:~/Development|⇒ git clone git@github.com:talentsparkio/CloudNote.git      21:56:10
Cloning into 'CloudNote'...
remote: Counting objects: 242, done.
remote: Compressing objects: 100% (154/154), done.
remote: Total 242 (delta 44), reused 242 (delta 44), pack-reused 0
Receiving objects: 100% (242/242), 9.43 MiB | 2.10 MiB/s, done.
Resolving deltas: 100% (44/44), done.
Checking connectivity... done.
vazexqi@daigo:~/Development|⇒ cd CloudNote
vazexqi@daigo:~/Development/CloudNote|⇒ git reset --hard InitialImport|      ± master ✓ 21:56:19
```

Remember to do `pod update`

Sets up CocoaPod, Objective-C Bridging and
NSAppTransportSecurity

PARSE



A screenshot of a web browser window showing the Parse.com application keys page. The URL in the address bar is `www.parse.com/apps/cloudnote--4/edit#keys`. The page title is "CloudNote" and the environment is "DEV". The left sidebar has a "General" section selected, followed by "Keys" (which is highlighted in blue), "Push", "Hosting", "Authentication", "Email", "Collaborators", and "Export data". The main content area is titled "Application Keys" and lists four key types: Application ID, Client Key, JavaScript Key, and .NET Key. Each key is represented by a text input field containing a long string of characters, with a blue selection bar on the right side of each field.

Key Type	Value
Application ID	DnAVErSXLQs1XHiPjm5tafWHs...
Client Key	QCRGgMt573gTUKHUODBpMaUXe...
JavaScript Key	TdufefmyVbvoXgLXBVsHduf68...
.NET Key	H0dtR8MQXJViosJTmsV5KpHDk...

Docs Billing Downloads Help Status

PARSE

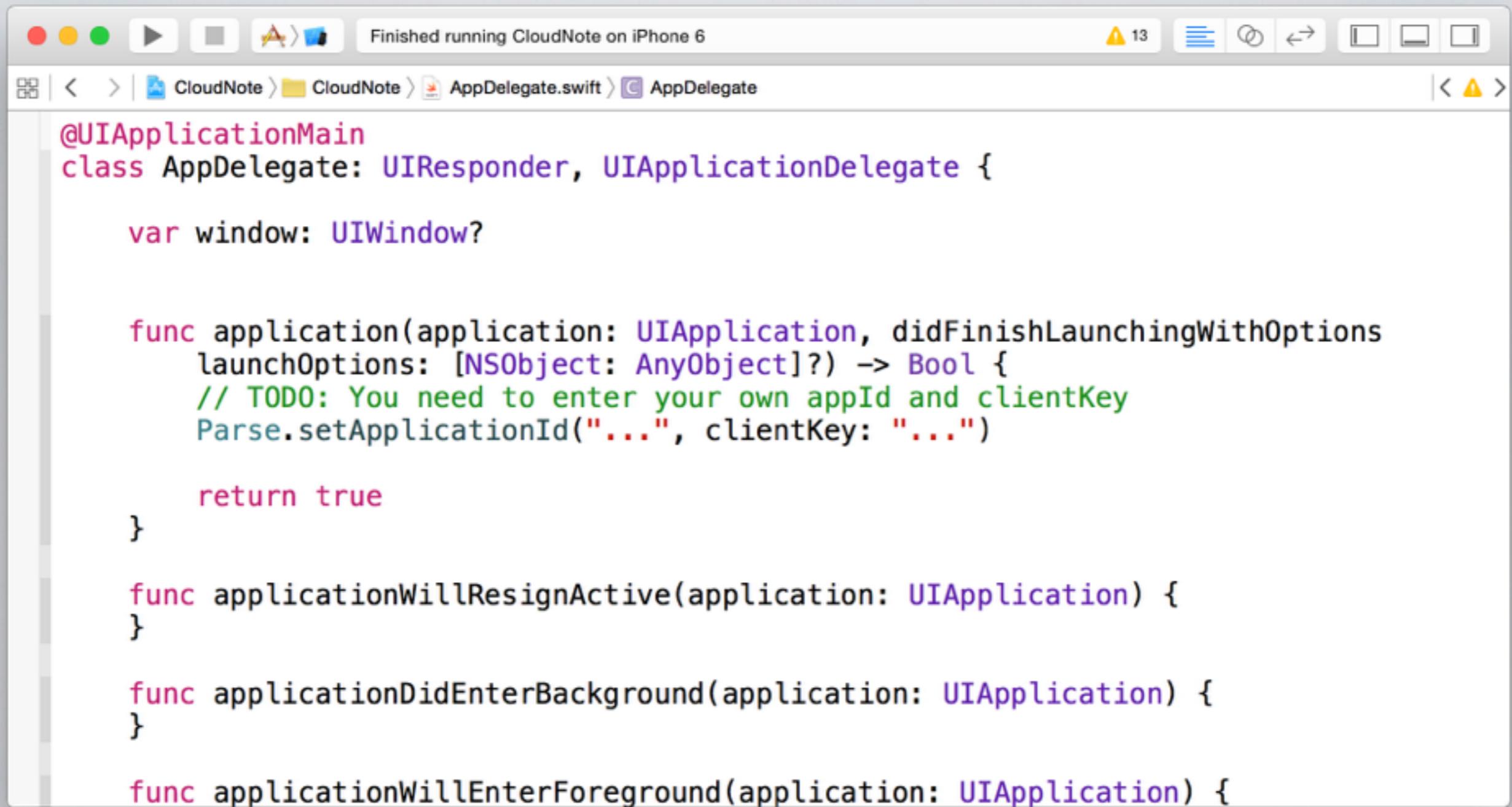
The screenshot shows the Parse.com application keys management interface. On the left, a sidebar menu includes General, Keys (which is selected and highlighted in blue), Push, Hosting, Authentication, Email, Collaborators, and Export data. The main content area is titled "Application Keys" and displays four key types: Application ID, Client Key, JavaScript Key, and .NET Key. A green callout box points to the Client Key and JavaScript Key fields, with the text "You need these two keys". The URL in the browser bar is www.parse.com/apps/cloudnote--4/edit#keys.

You need these two keys

Key Type	Value
Client Key	TdufefmyVbvoXgLXBVsHduf68...
JavaScript Key	TdufefmyVbvoXgLXBVsHduf68...
.NET Key	H0dtR8MQXJViosJTmsV5KpHDk...

Docs Billing Downloads Help Status

SKELETON CODE



A screenshot of the Xcode IDE interface. The title bar shows "Finished running CloudNote on iPhone 6". The navigation bar displays the file path: "CloudNote > CloudNote > AppDelegate.swift > AppDelegate". The main editor area contains the following Swift code:

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(application: UIApplication, didFinishLaunchingWithOptions
        launchOptions: [NSObject: AnyObject]?) -> Bool {
        // TODO: You need to enter your own appId and clientKey
        Parse.setApplicationId("...", clientKey: "...")

        return true
    }

    func applicationWillResignActive(application: UIApplication) {

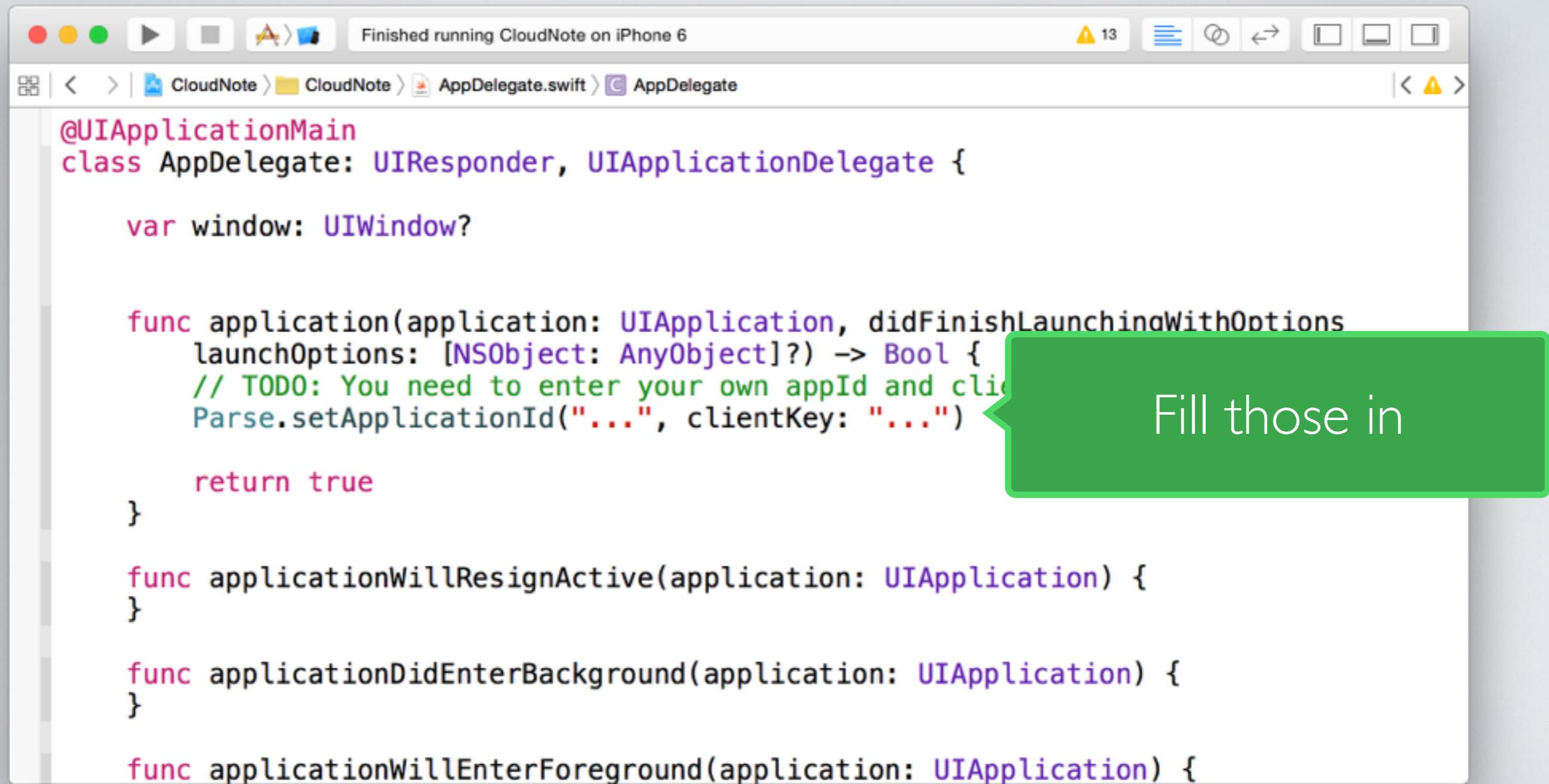
    }

    func applicationDidEnterBackground(application: UIApplication) {

    }

    func applicationWillEnterForeground(application: UIApplication) {
```

SKELETON CODE



Finished running CloudNote on iPhone 6

CloudNote > CloudNote > AppDelegate.swift > AppDelegate

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(application: UIApplication, didFinishLaunchingWithOptions
        launchOptions: [NSObject: AnyObject]?) -> Bool {
        // TODO: You need to enter your own appId and client key
        Parse.setApplicationId("...", clientKey: "...")

        return true
    }

    func applicationWillResignActive(application: UIApplication) {

    }

    func applicationDidEnterBackground(application: UIApplication) {

    }

    func applicationWillEnterForeground(application: UIApplication) {
```

Fill those in

PARSE

The more you know.

Explore our comprehensive guides for each platform to get started. Or, jump right in with our detailed API references and tutorials.



iOS
v1.7.5

Guide >

API Reference >

Tutorials >

DOWNLOAD



Android
v1.9.4

Guide >

API Reference >

Tutorials >

DOWNLOAD

PARSE

The screenshot shows a web browser window with the URL www.parse.com/docs in the address bar. The page features a large heading "PARSE" at the top, followed by a sub-headline "The more you know." Below this, there's a brief description encouraging users to explore comprehensive guides or jump right into API references and tutorials. Two main sections are displayed: one for iOS (version v1.7.5) and one for Android (version v1.9.4). Each section includes links for "Guide", "API Reference", and "Tutorials", and a prominent blue "DOWNLOAD" button. A green call-to-action button with the text "Access the documentation" is overlaid on the Android section.

The more you know.

Explore our comprehensive guides for each platform to get started. Or, jump right in with our detailed API references and tutorials.

iOS
v1.7.5

Guide

API Reference

Tutorials

DOWNLOAD

Android
v1.9.4

Access the documentation

Red Yellow Green Back Forward Home Stop Refresh More Apple Inc. developer.apple.com/library/ios/Developer/

iOS Developer Library – Prerelease

Start Developing iOS Apps (Swift)

Search

Getting Started

- Jump Right In**
- Learn the Essentials of Swift

Building the UI

- Build a Basic UI
- Connect the UI to Code
- Work with View Controllers
- Implement a Custom Control
- Define Your Data Model

Working with Table Views

- Create a Table View
- Implement Navigation

Jump Right In

IMPORTANT

This is a preliminary document for an API or technology in development. Apple is supplying this information to you in order to solicit your feedback. Please do not rely on this information as a final product, as it may change significantly before its final release. You should plan for the adoption of the technologies and programming interfaces described herein for use or reference when developing your application, with the understanding that their final values and forms will be determined at the time of their final implementation. This information is subject to change, and software implemented according to this document may be tested with final operating system software and final documentation. Newer versions of this document are provided with future betas of the API or technology.

Start Developing iOS Apps (Swift) is the perfect starting point for creating apps that run on iOS. View this set of incremental lessons as a guided introduction to building your first app—including major concepts, and best practices that will ease your path.

Each lesson contains a tutorial and the conceptual information you need to complete it. The

Red Yellow Green Backspace Home Stop Refresh Forward Search

Apple Inc. developer.apple.com/library

iOS Developer Library – Prerelease

Start Developing iOS Apps (Swift)

Getting Started

- Jump Right In
- Learn the Essentials of Swift

Building the UI

- Build a Basic UI
- Connect the UI to Code
- Work with View Controllers
- Implement a Custom Control
- Define Your App's Data Model

Working with Data

- Create a Core Data Model
- Implement Fetch Requests

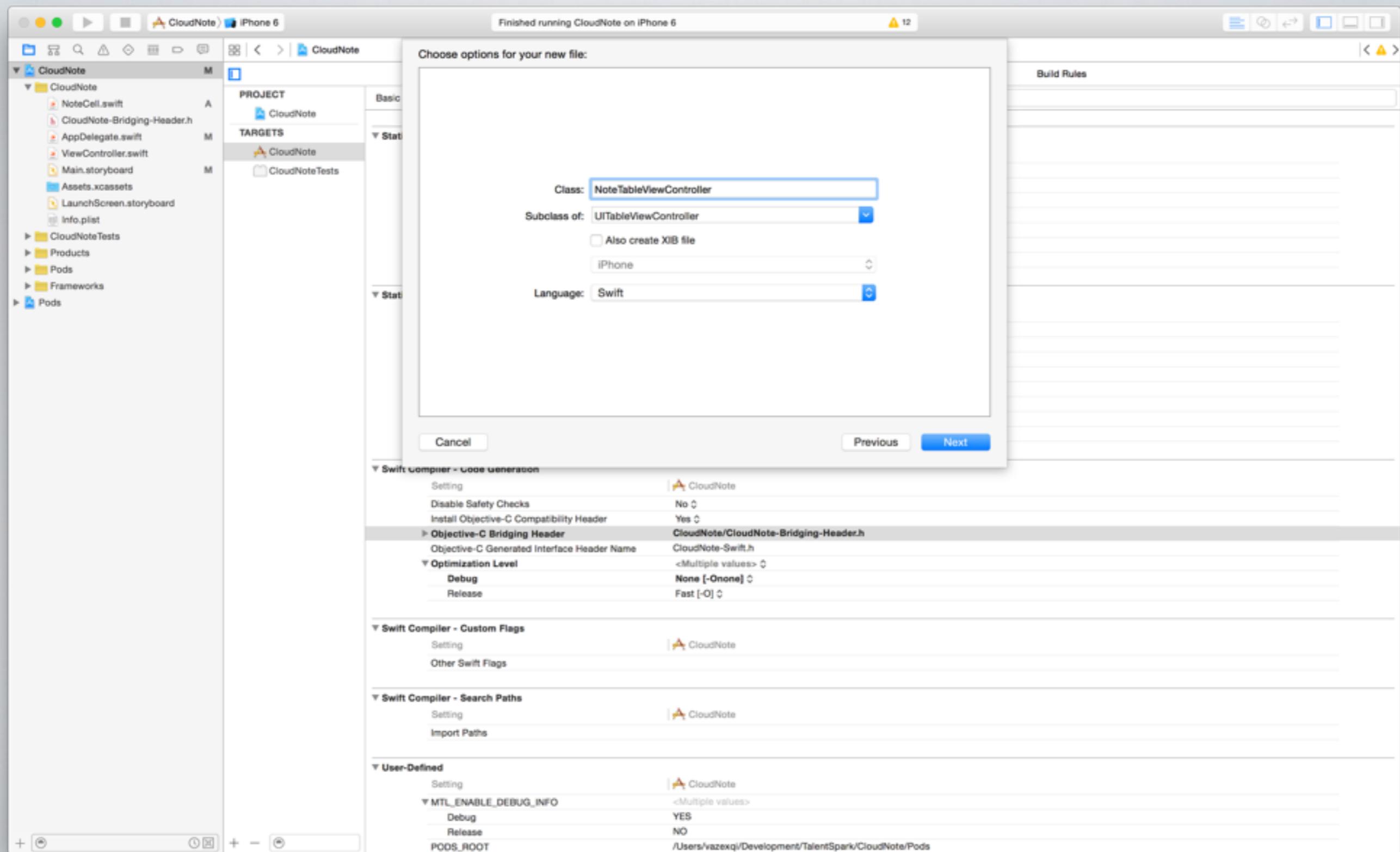
Jump Right In

IMPORTANT

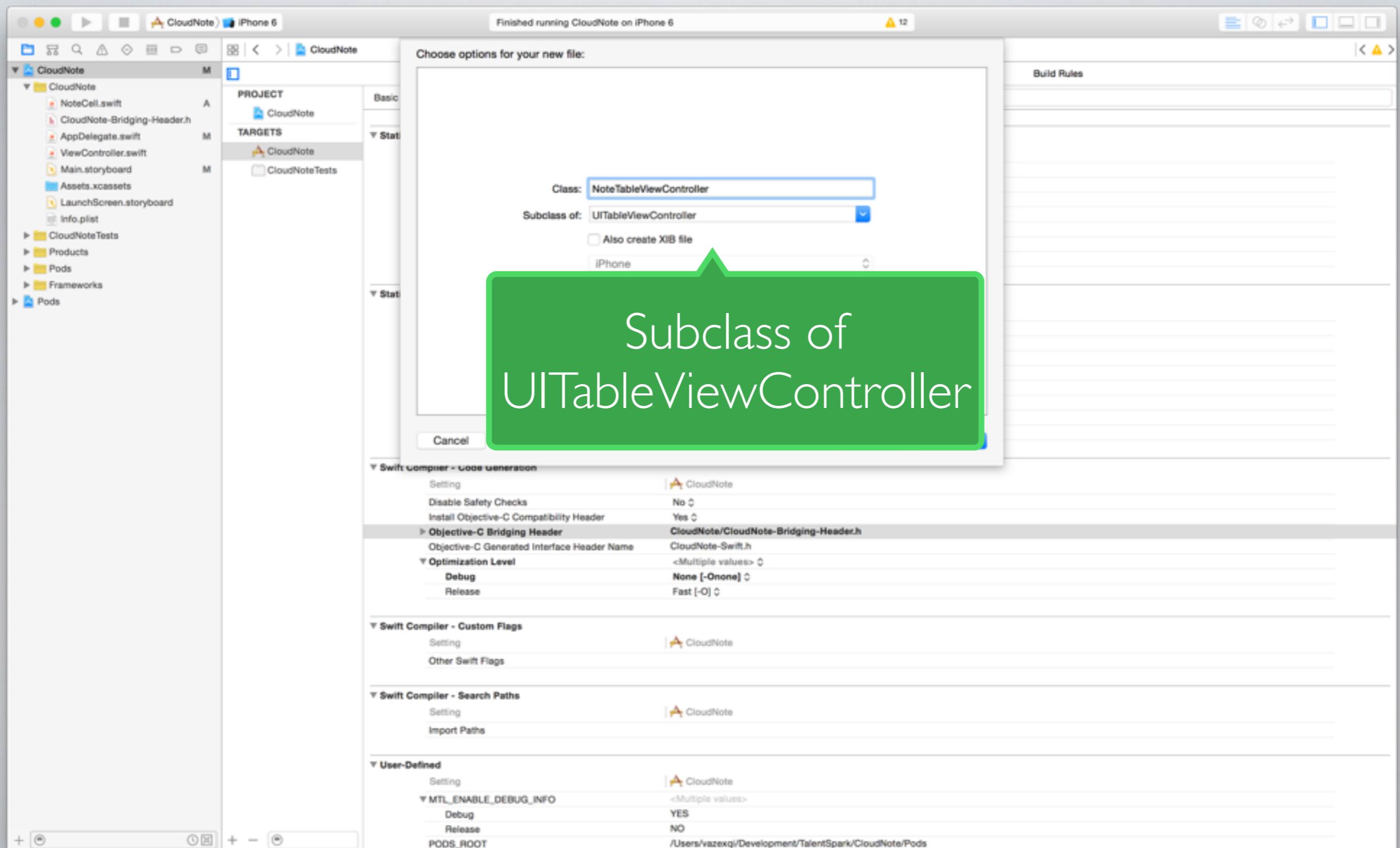
This is a preliminary document for an API or technology in development. Apple is supplying this information to you as part of its commitment to open source. It is for informational purposes only and is not a contract. You plan for the adoption of the technologies and programming interfaces described herein for use or sale in your products. This information is subject to change, and software implemented according to this document may not be fully tested or supported by Apple. This document is not intended to be used with final operating system software and final documentation. Newer versions of this document will be provided with future betas of the API or technology.

This is tricky stuff. I've purposely followed this write-up as closely as possible so you can re-read it. I've only changed the backend

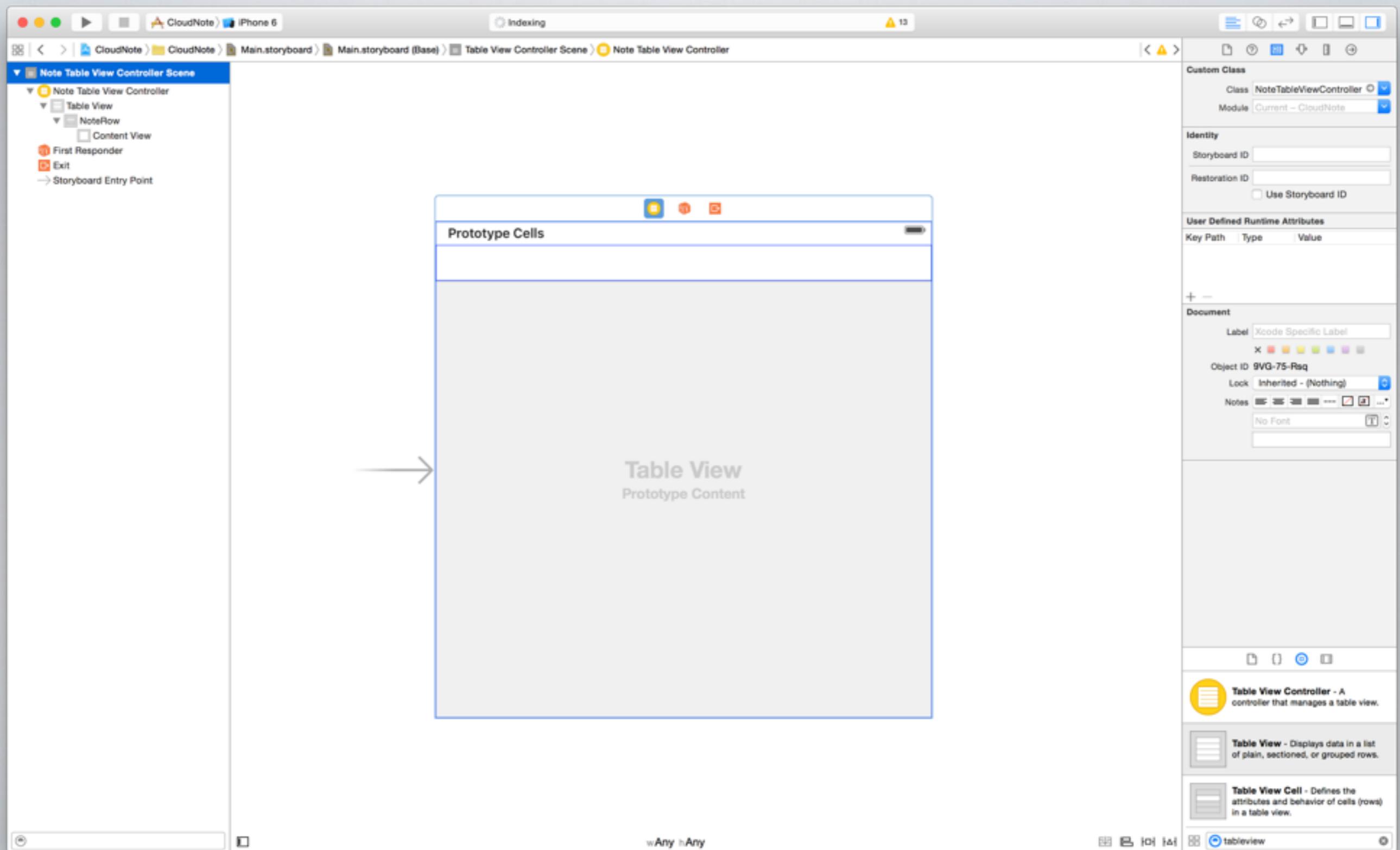
TABLEVIEWCONTROLLER



TABLEVIEWCONTROLLER

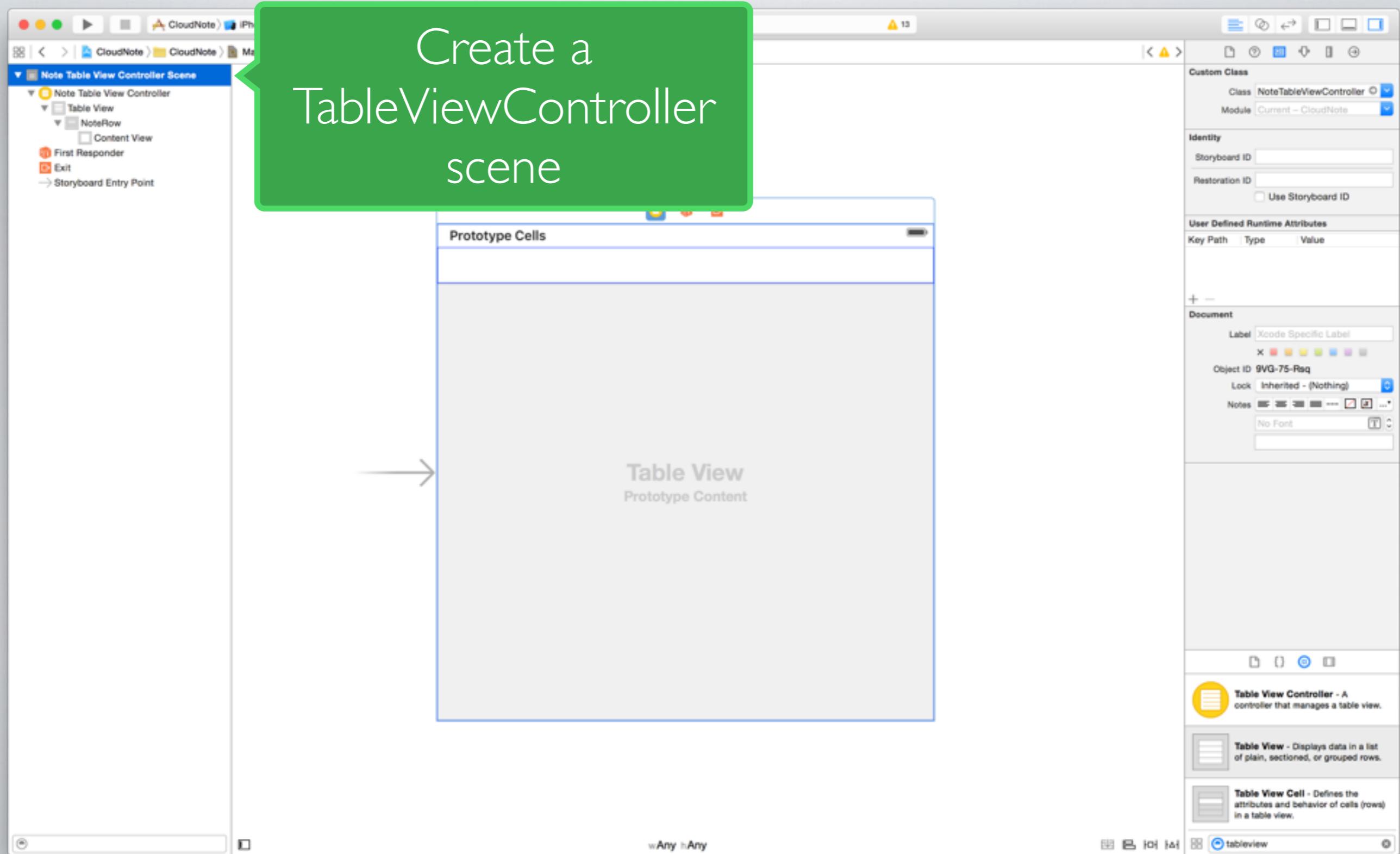


TABLEVIEWCONTROLLER



TABLEVIEWCONTROLLER

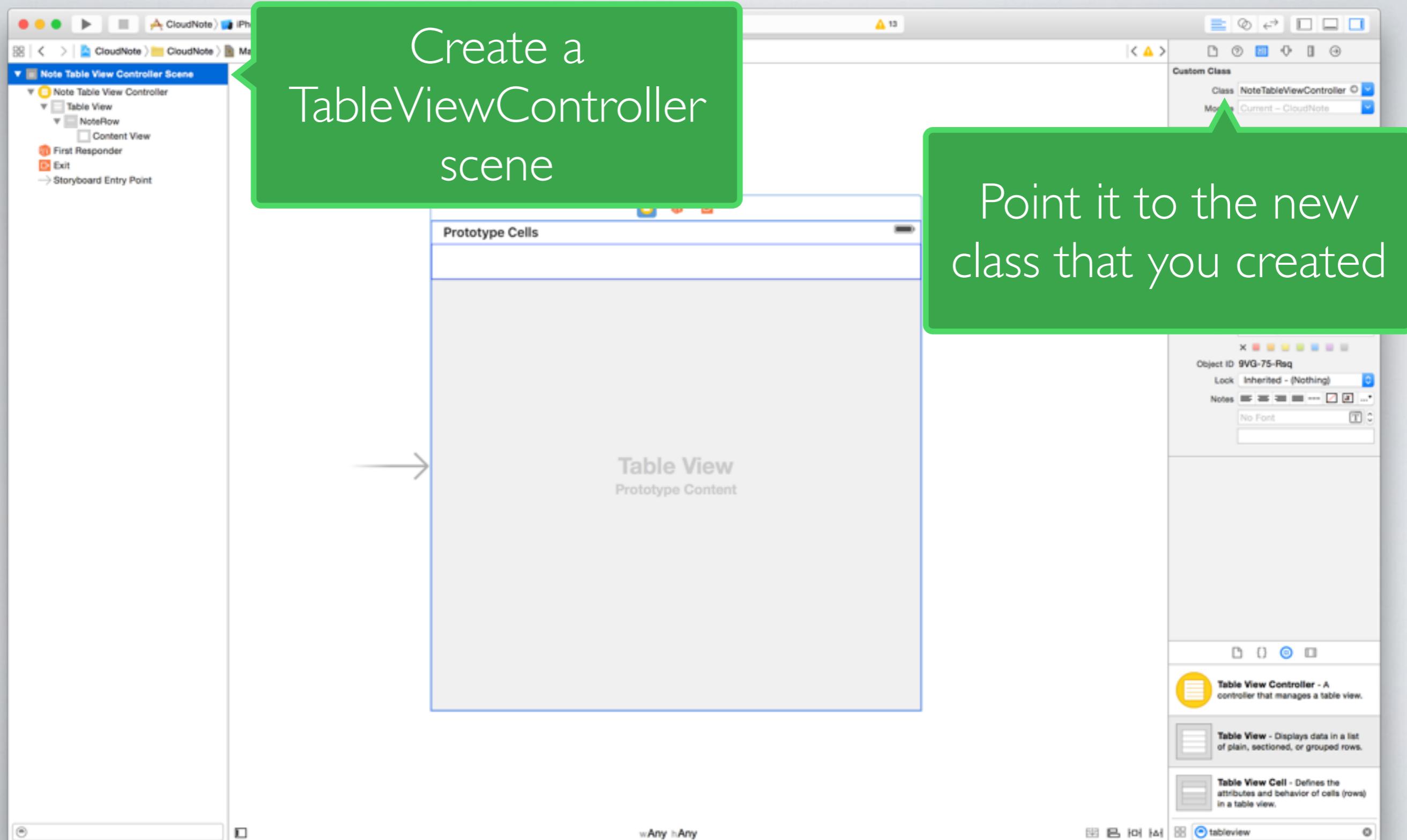
Create a
TableViewController
scene



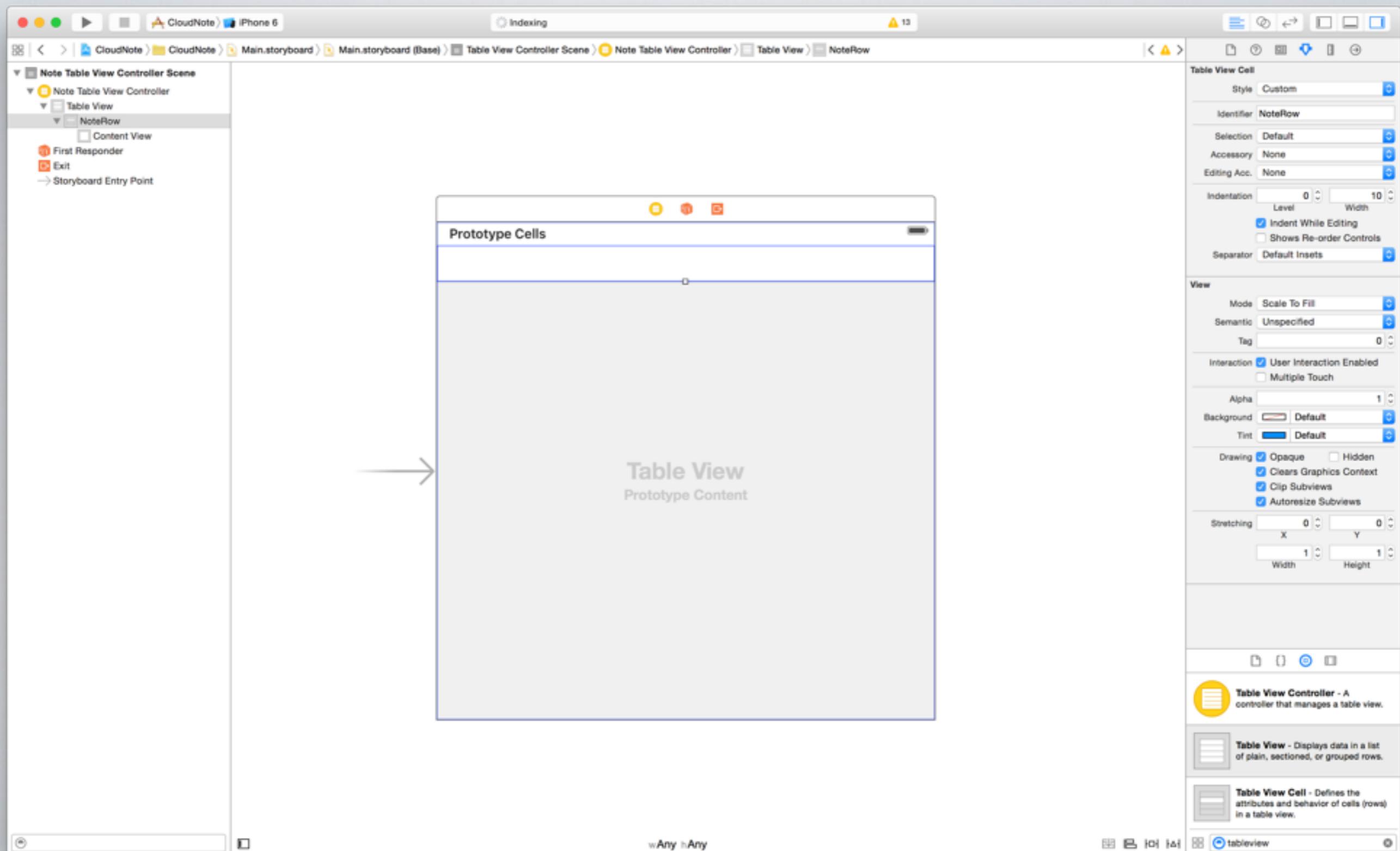
TABLEVIEWCONTROLLER

Create a
TableViewController
scene

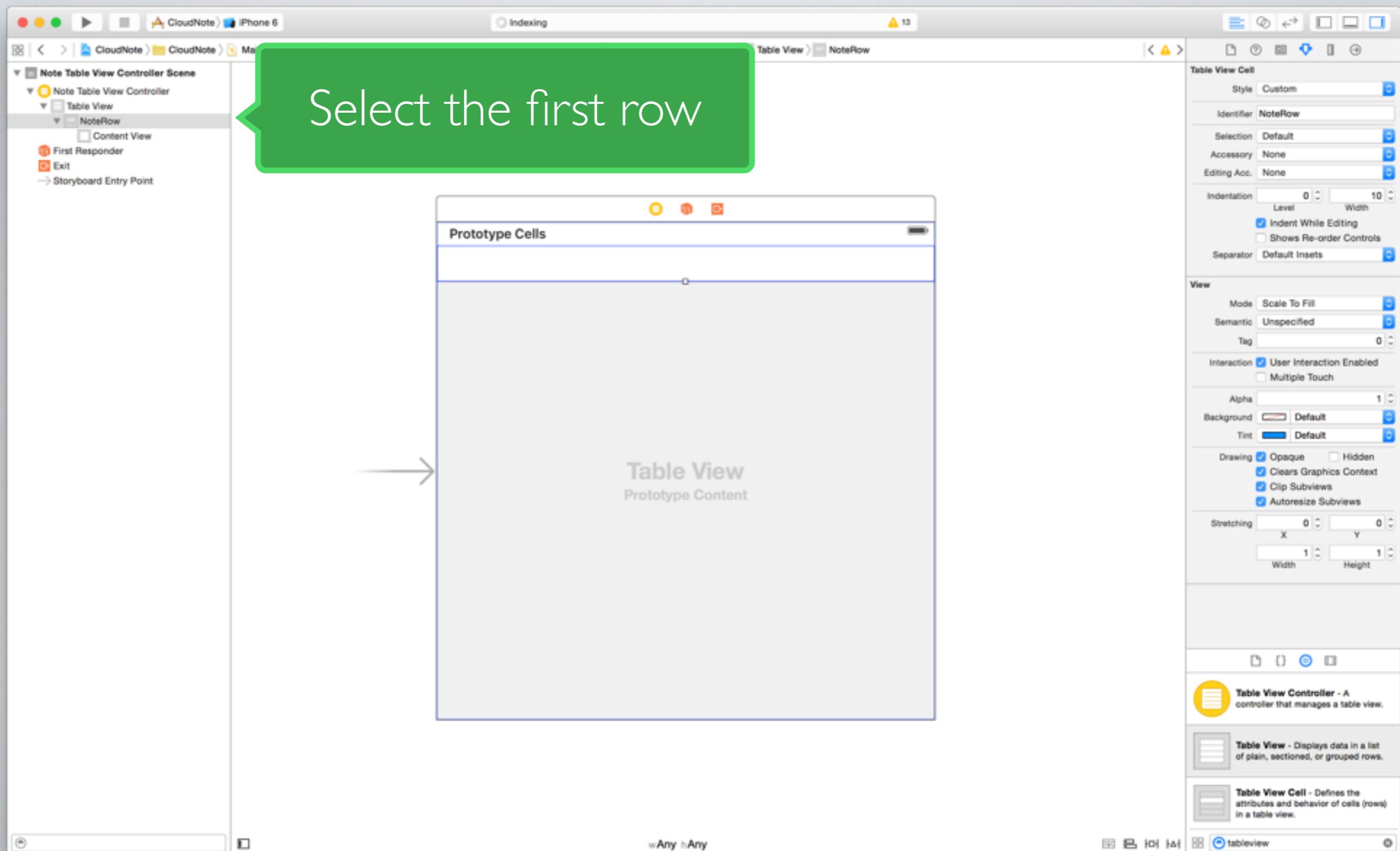
Point it to the new
class that you created



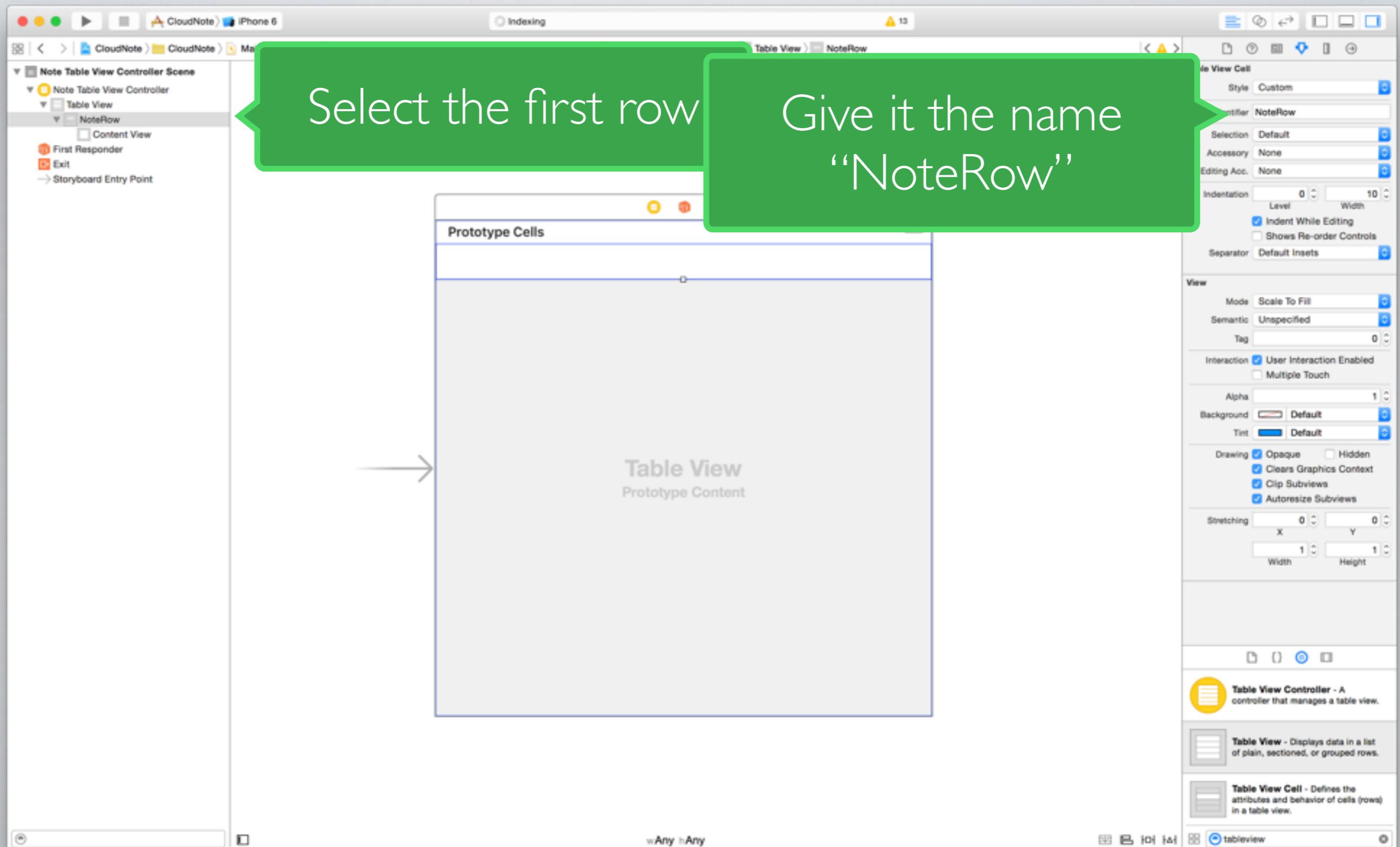
TABLEVIEWCONTROLLER



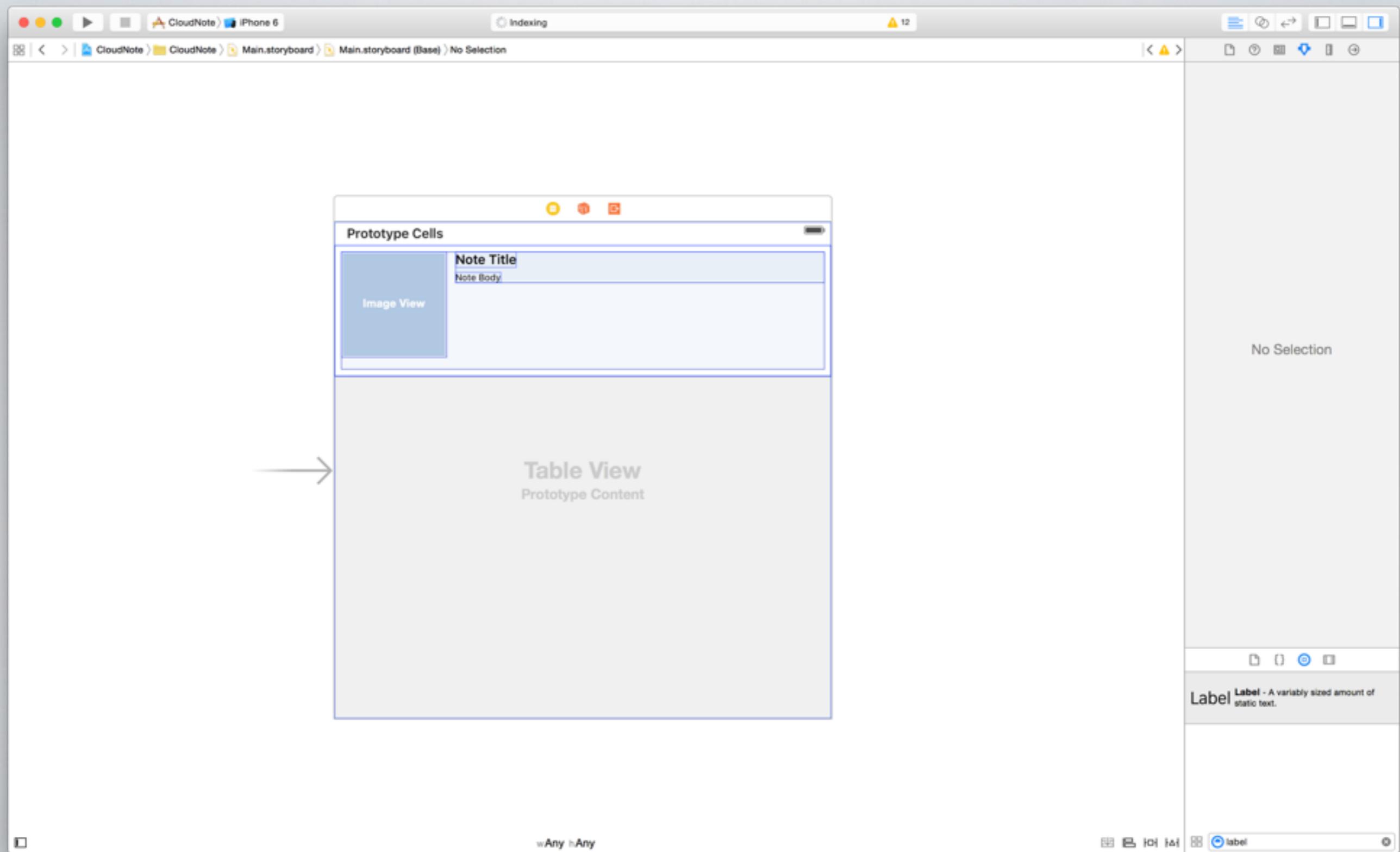
TABLEVIEWCONTROLLER



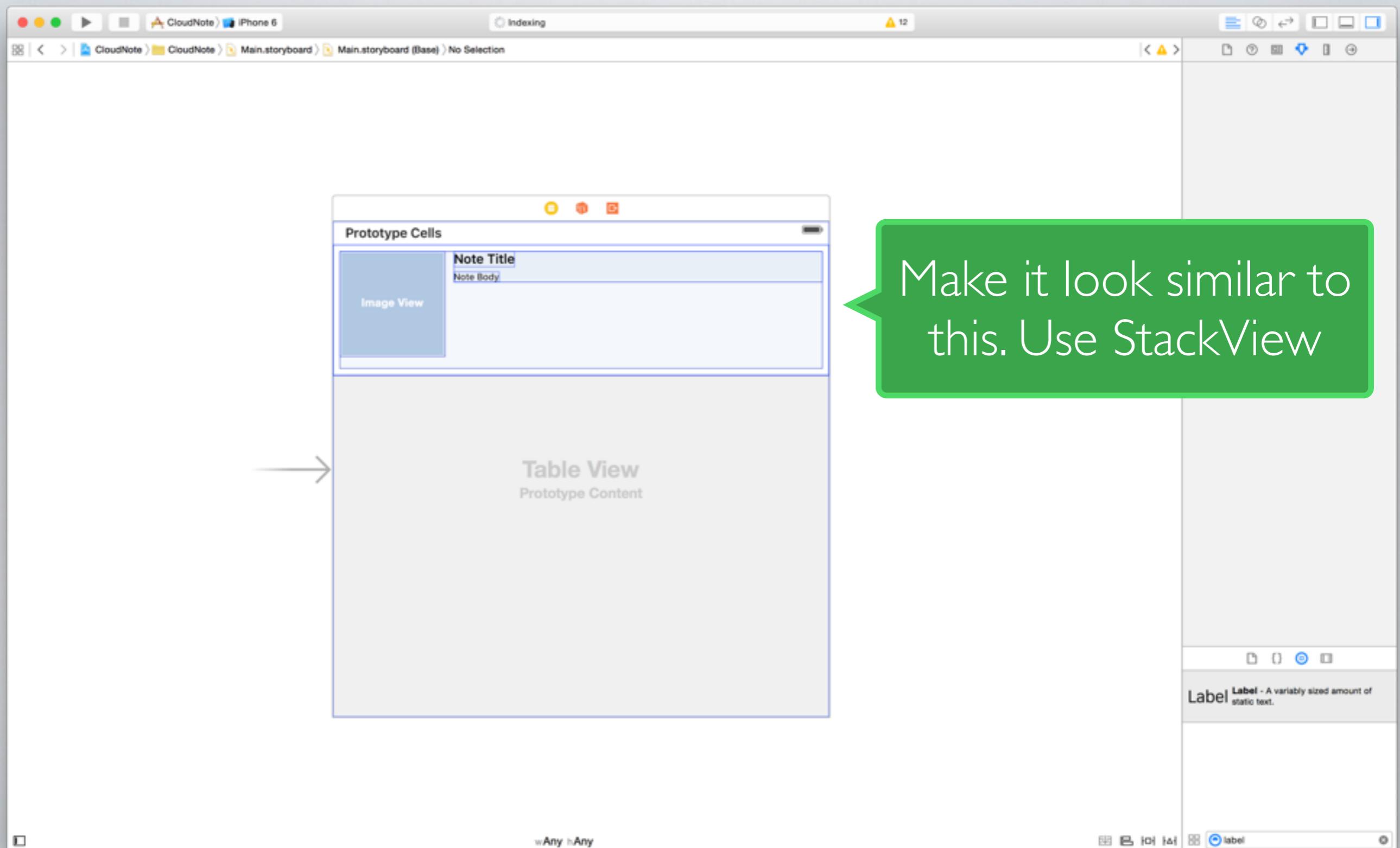
TABLEVIEWCONTROLLER



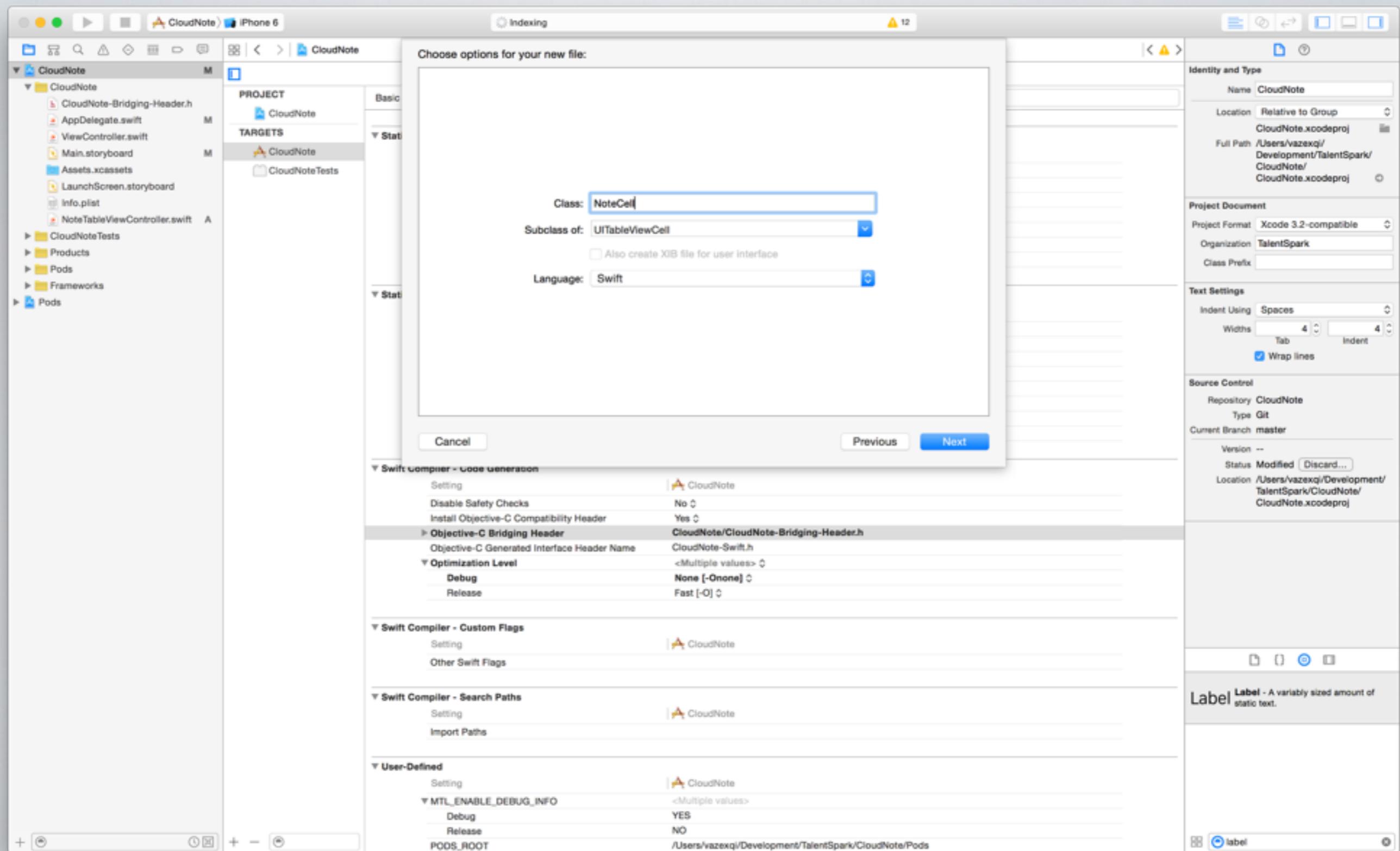
TABLEVIEWCONTROLLER



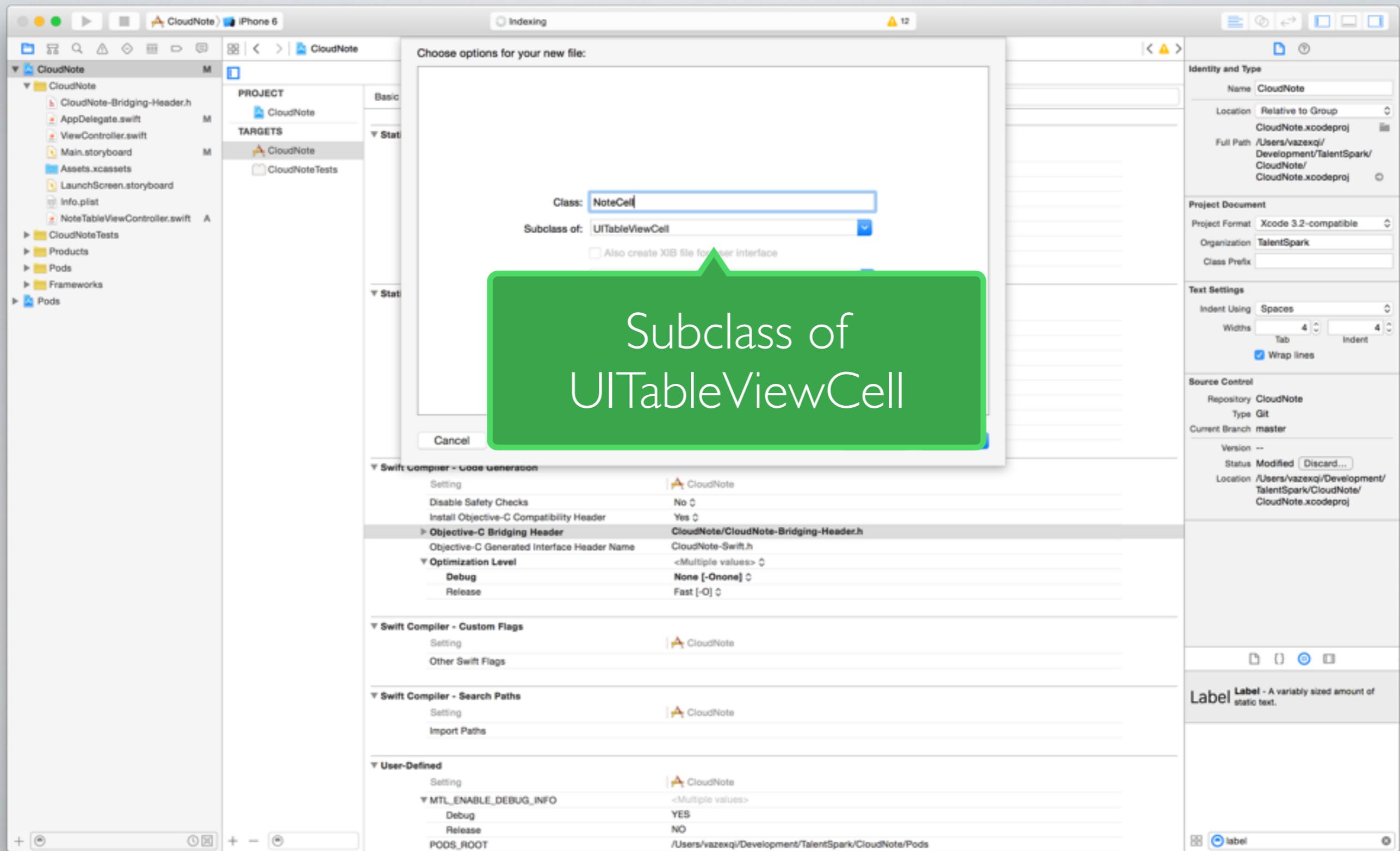
TABLEVIEWCONTROLLER



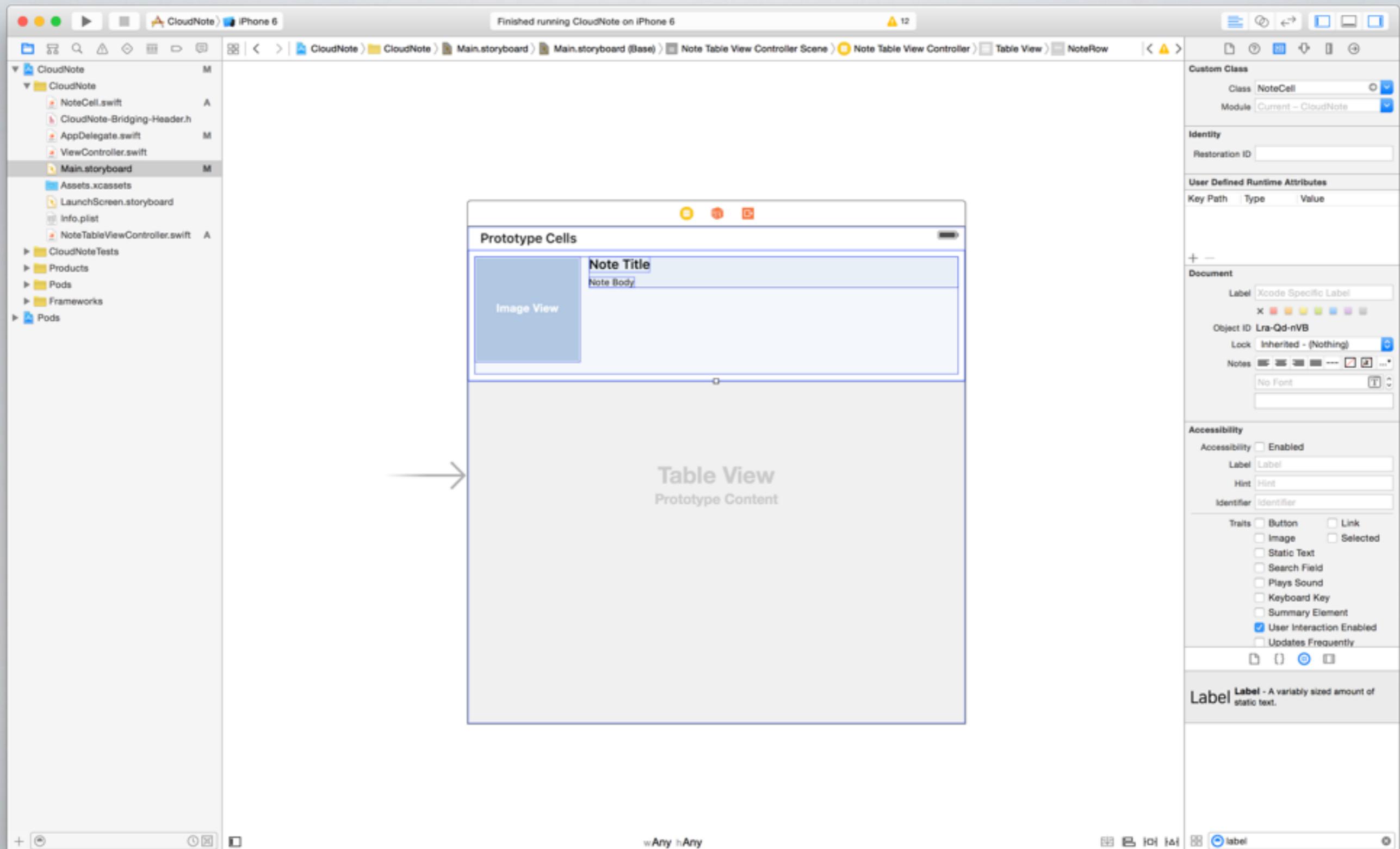
CUSTOM TABLEVIEWCELL



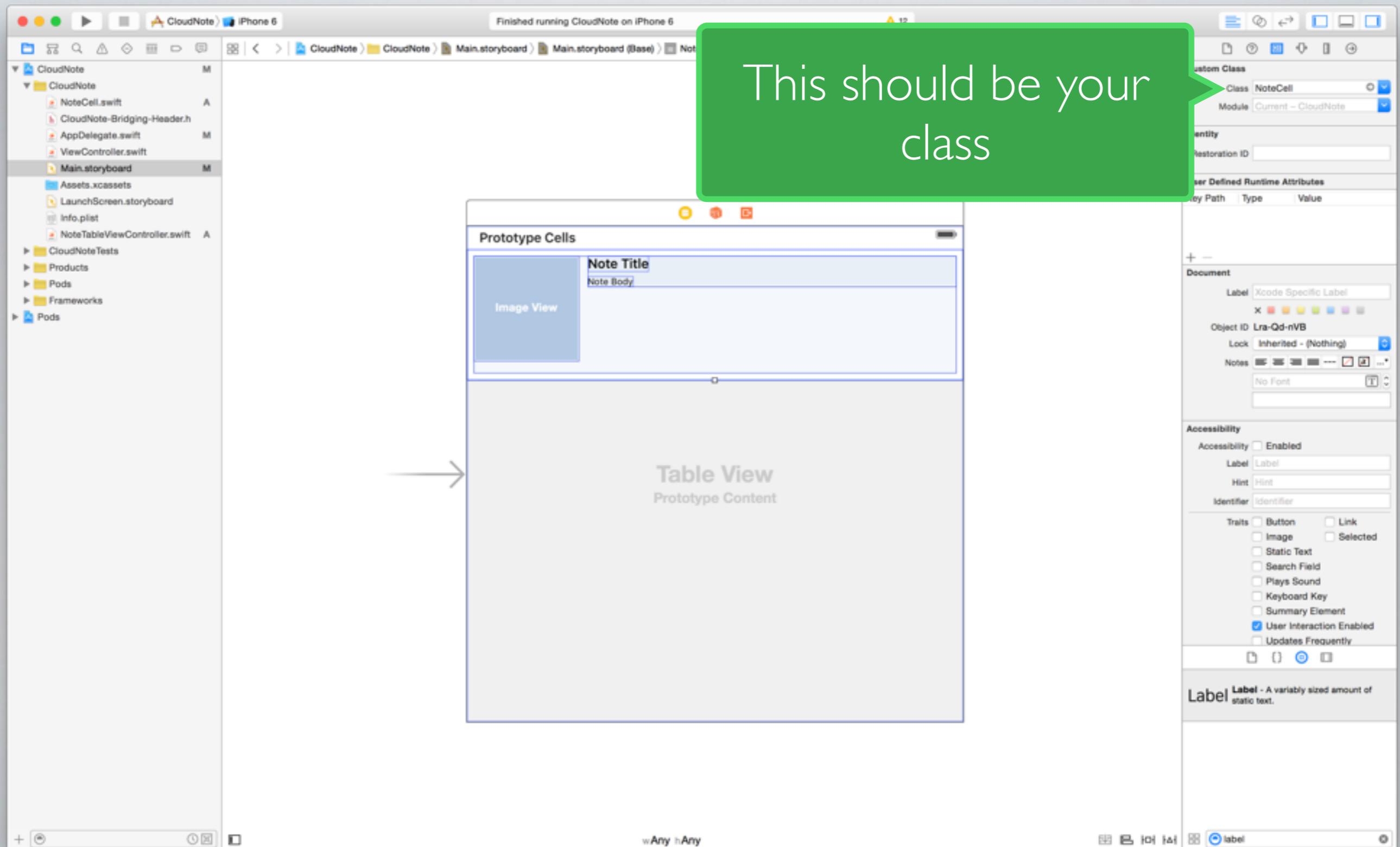
CUSTOM TABLEVIEWCELL



CUSTOM TABLEVIEWCELL



CUSTOM TABLEVIEWCELL



CUSTOM TABLEVIEWCELL

The screenshot shows the Xcode interface with a storyboard and a code editor side-by-side.

Storyboard (Left): A "Table View" prototype content is displayed. It contains a "Prototype Cells" section with a single cell. The cell has three components: an "Image View" (blue square), a "Note Title" label (text "Note Title" above a line), and a "Note Body" label (text "Note Body" below a line).

Code Editor (Right): The file is "NoteCell.swift" under the "CloudNote" project. The code defines a custom UITableViewCell:

```
// NoteCell.swift
// CloudNote
//
// Created by Nick Chen on 8/8/15.
// Copyright © 2015 TalentSpark. All rights reserved.

import UIKit

class NoteCell: UITableViewCell {

    @IBOutlet weak var notePhoto: UIImageView!
    @IBOutlet weak var noteTitle: UILabel!
    @IBOutlet weak var noteBody: UILabel!

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }

    override func setSelected(selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)
        // Configure the view for the selected state
    }
}
```

CUSTOM TABLEVIEWCELL

The screenshot shows the Xcode interface with a project named "CloudNote" running on an iPhone 6 simulator. The main area displays the storyboard with a prototype cell containing an image view and two labels. To the right, the "NoteCell.swift" file is open, showing the implementation of the `NoteCell` class. A green callout box with the text "Ctrl + Drag and form IBOutlets" points from the storyboard to the code, specifically to the line where three outlets are declared: `notePhoto`, `noteTitle`, and `noteBody`. The code also includes methods for `awakeFromNib` and `setSelected`.

```
// NoteCell.swift
// CloudNote
//
// Created by Nick Chen on 8/8/15.
// Copyright © 2015 TalentSpark. All rights reserved.

import UIKit

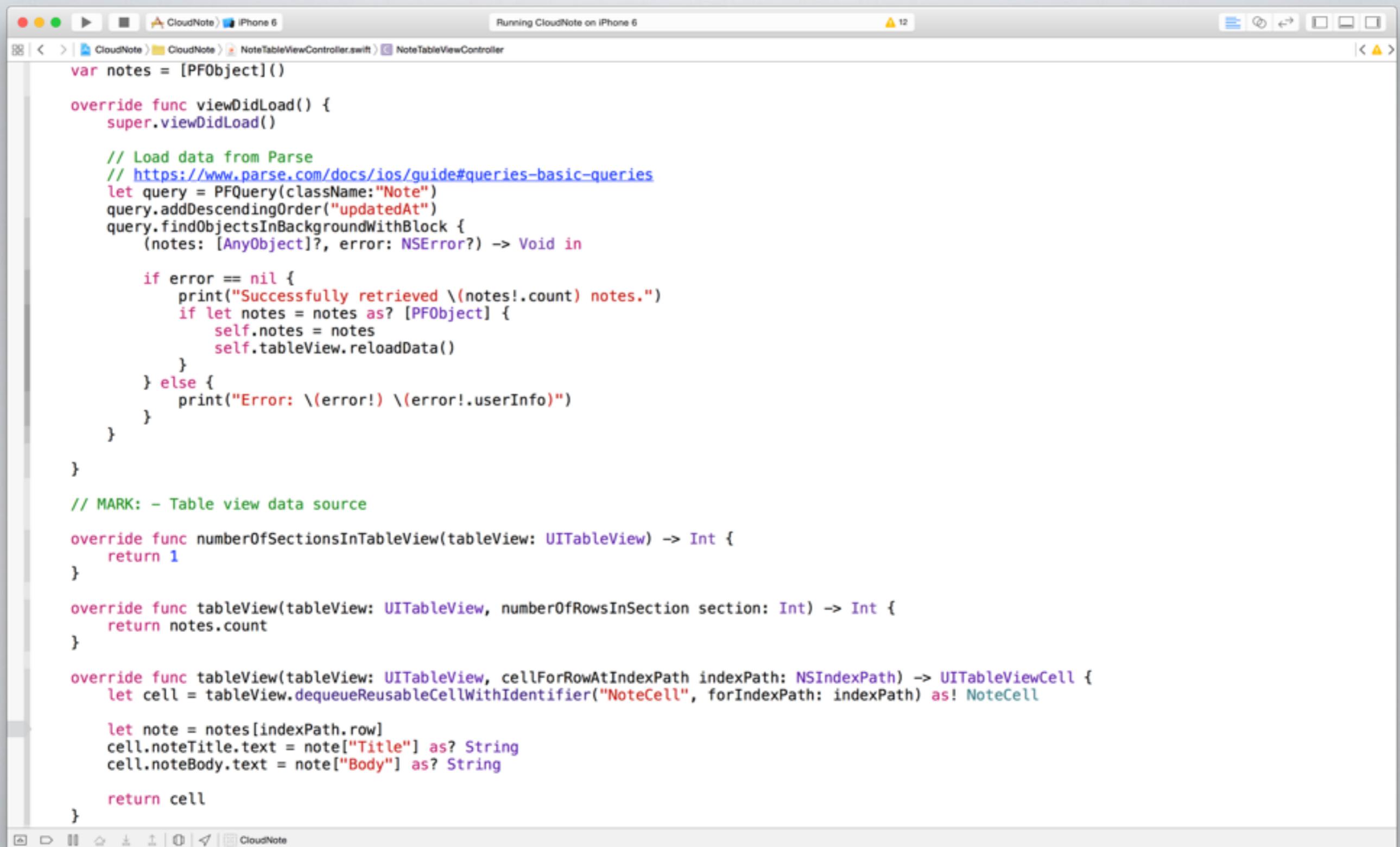
class NoteCell: UITableViewCell {

    @IBOutlet weak var notePhoto: UIImageView!
    @IBOutlet weak var noteTitle: UILabel!
    @IBOutlet weak var noteBody: UILabel!

    override func awakeFromNib() {
        super.awakeFromNib()
        // code
    }

    func setSelected(selected: Bool, animated: Bool) {
        (selected, animated)
        // view for the selected state
    }
}
```

TABLEVIEWCONTROLLER



The screenshot shows the Xcode IDE interface with the following details:

- Title Bar:** "CloudNote" and "iPhone 6".
- Toolbar:** Standard Xcode toolbar.
- Document Outline:** Shows "CloudNote" and "NoteTableViewController.swift".
- Text Editor:** Displays the Swift code for NoteTableViewController. The code handles data retrieval from Parse, sets up the table view data source, and configures cells.
- Top Status Bar:** Shows "Running CloudNote on iPhone 6" and a warning icon with "12".
- Bottom Status Bar:** Shows standard iOS status icons.

```
var notes = [PFObject]()

override func viewDidLoad() {
    super.viewDidLoad()

    // Load data from Parse
    // https://www.parse.com/docs/ios/guide#queries-basic-queries
    let query = PFQuery(className:"Note")
    query.addDescendingOrder("updatedAt")
    query.findObjectsInBackgroundWithBlock {
        (notes: [AnyObject]?, error: NSError?) -> Void in

        if error == nil {
            print("Successfully retrieved \(notes!.count) notes.")
            if let notes = notes as? [PFObject] {
                self.notes = notes
                self.tableView.reloadData()
            }
        } else {
            print("Error: \(error!) \(error!.userInfo)")
        }
    }
}

// MARK: - Table view data source

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {
    return 1
}

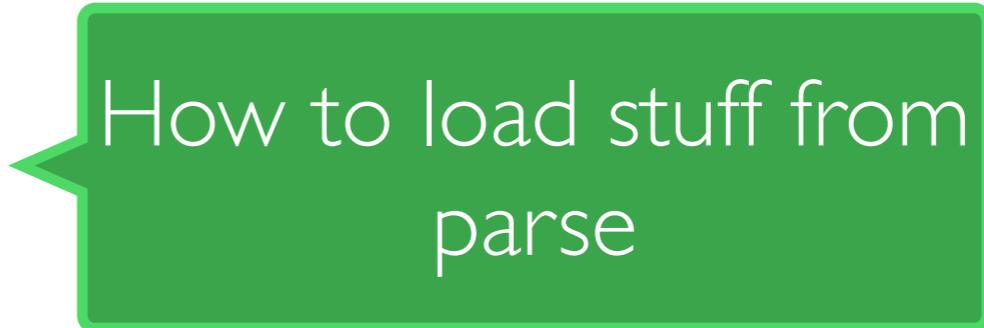
override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return notes.count
}

override func tableView(tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "NoteCell", for: indexPath) as! NoteCell

    let note = notes[indexPath.row]
    cell.noteTitle.text = note["Title"] as? String
    cell.noteBody.text = note["Body"] as? String

    return cell
}
```

TABLEVIEWCONTROLLER



Running CloudNote on iPhone 6

```
var notes = [PFObject]()

override func viewDidLoad() {
    super.viewDidLoad()

    // Load data from Parse
    // https://www.parse.com/docs/ios/guide#queries-basic-queries
    let query = PFQuery(className:"Note")
    query.addDescendingOrder("updatedAt")
    query.findObjectsInBackgroundWithBlock {
        (notes: [AnyObject]?, error: NSError?) -> Void in

        if error == nil {
            print("Successfully retrieved \(notes!.count) notes.")
            if let notes = notes as? [PFObject] {
                self.notes = notes
                self.tableView.reloadData()
            }
        } else {
            print("Error: \(error!) \(error!.userInfo)")
        }
    }
}

// MARK: - Table view data source

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {
    return 1
}

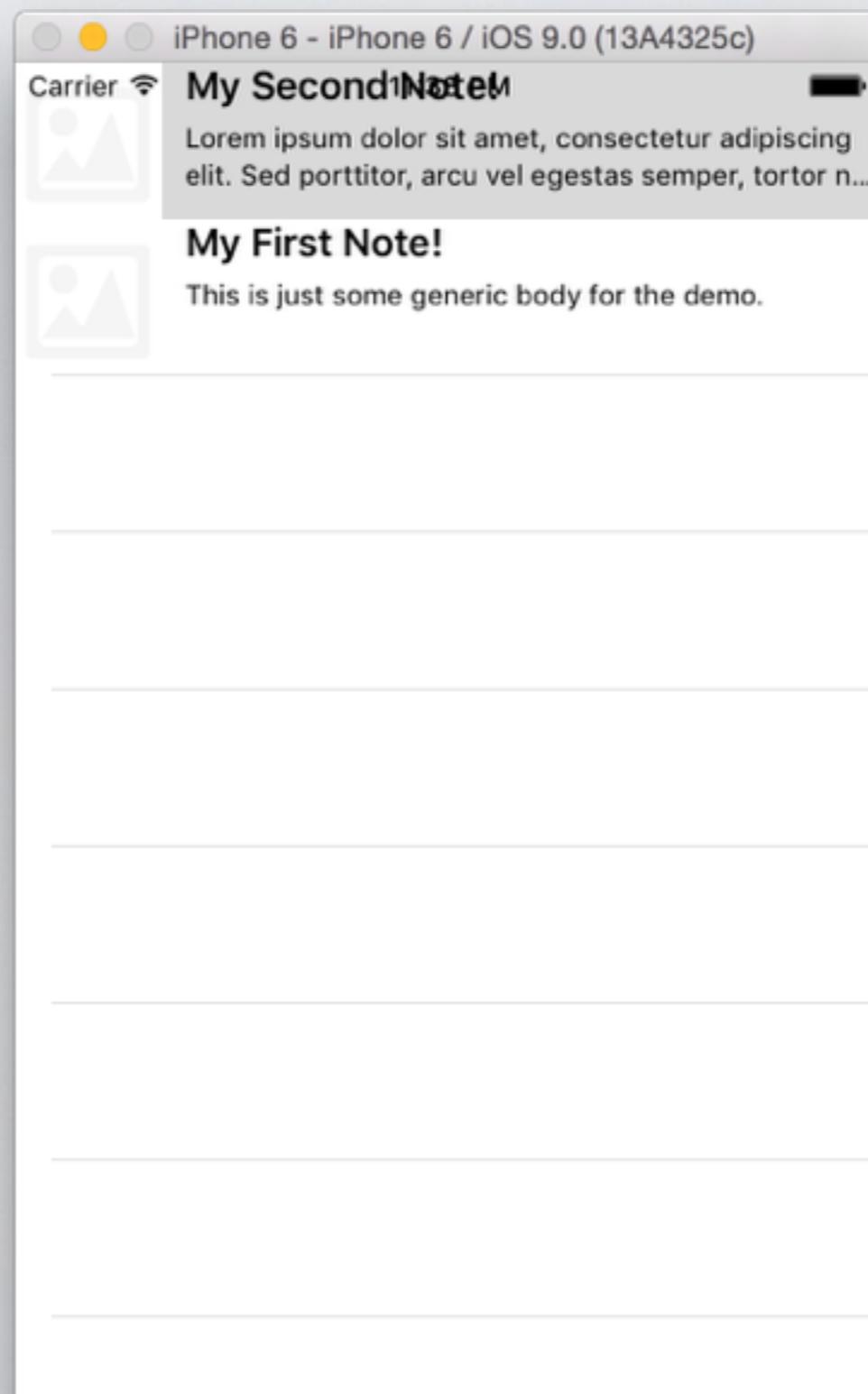
override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return notes.count
}

override func tableView(tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier:"NoteCell", for:indexPath) as! NoteCell

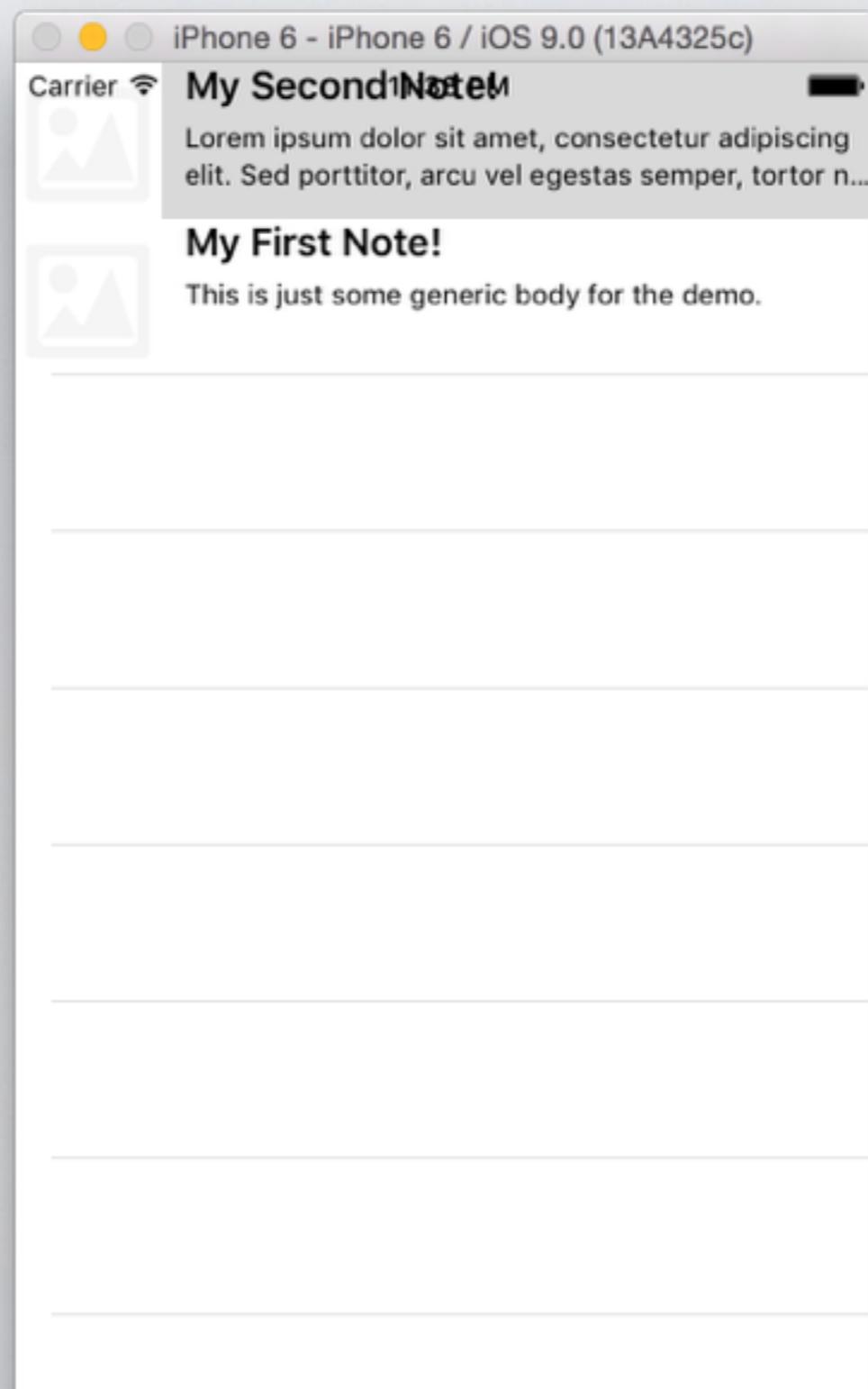
    let note = notes[indexPath.row]
    cell.noteTitle.text = note["Title"] as? String
    cell.noteBody.text = note["Body"] as? String

    return cell
}
```

CURRENTVIEW

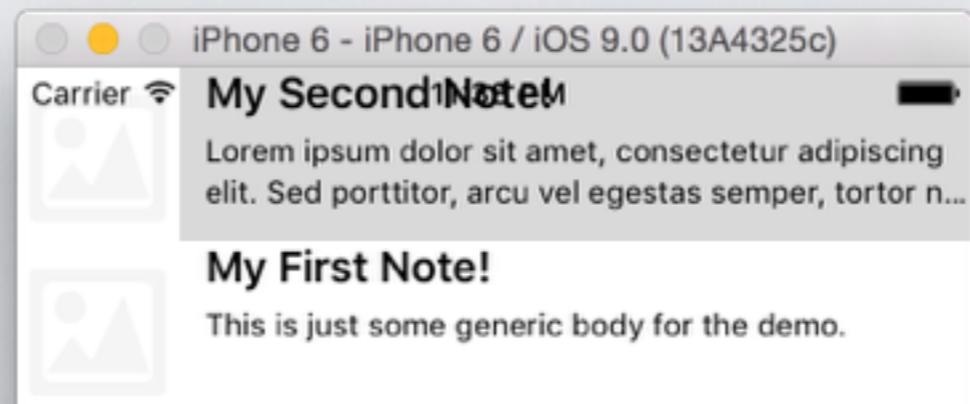


CURRENT VIEW



Yes, it bleeds
onto the top
layer.

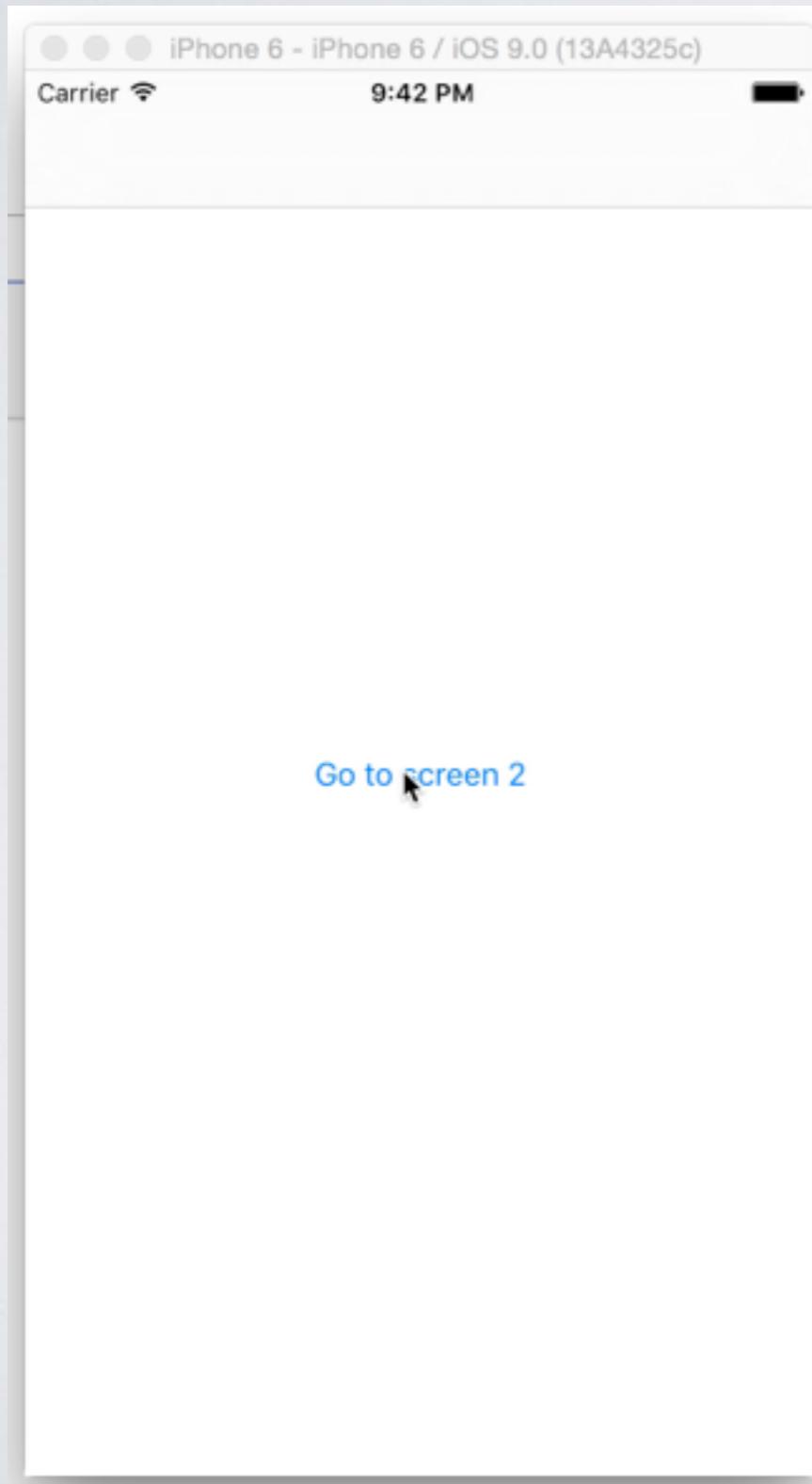
CURRENT VIEW



Yes, it bleeds
onto the top
layer.

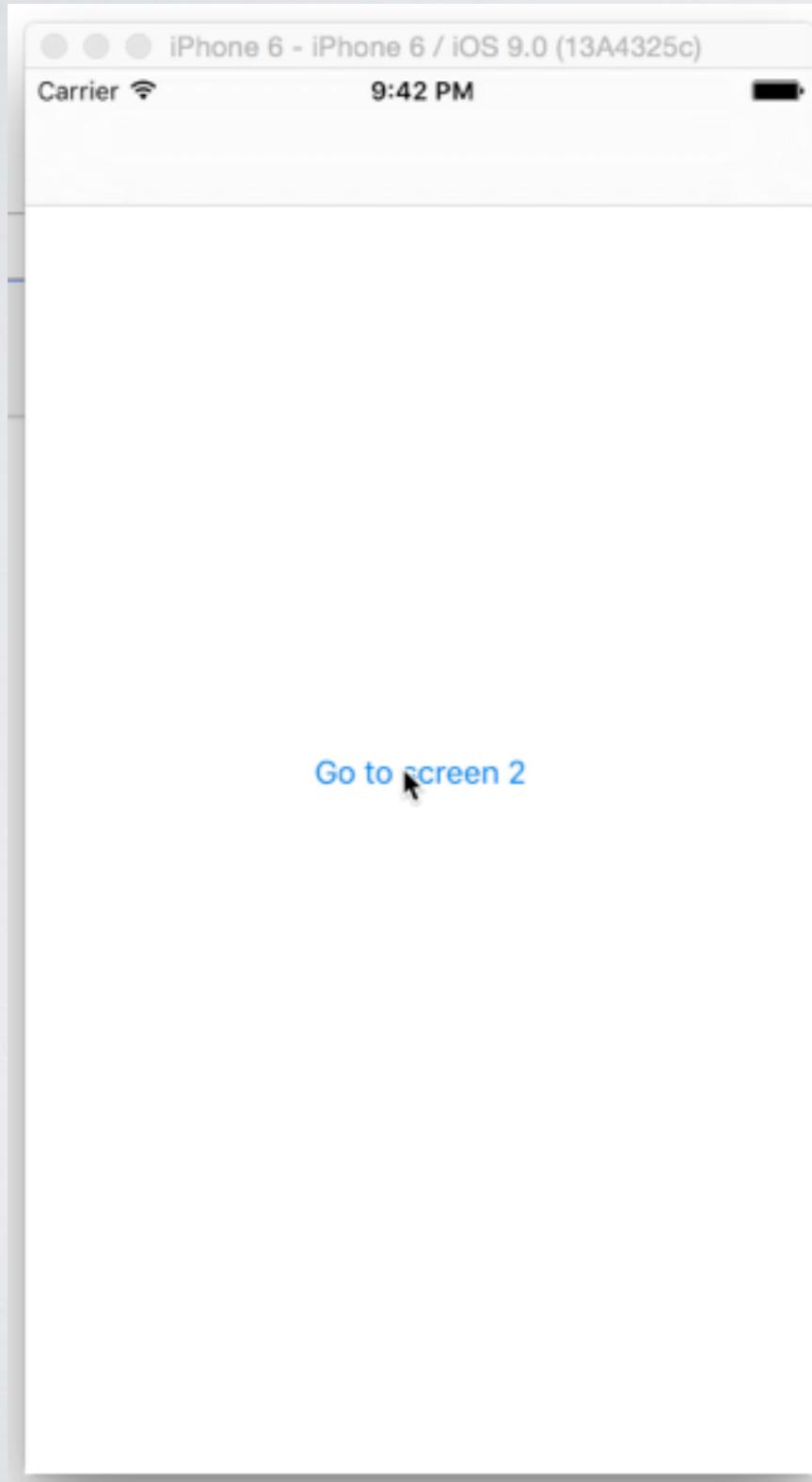
Let's fix this using
NavigationController

NAVIGATIONCONTROLLER



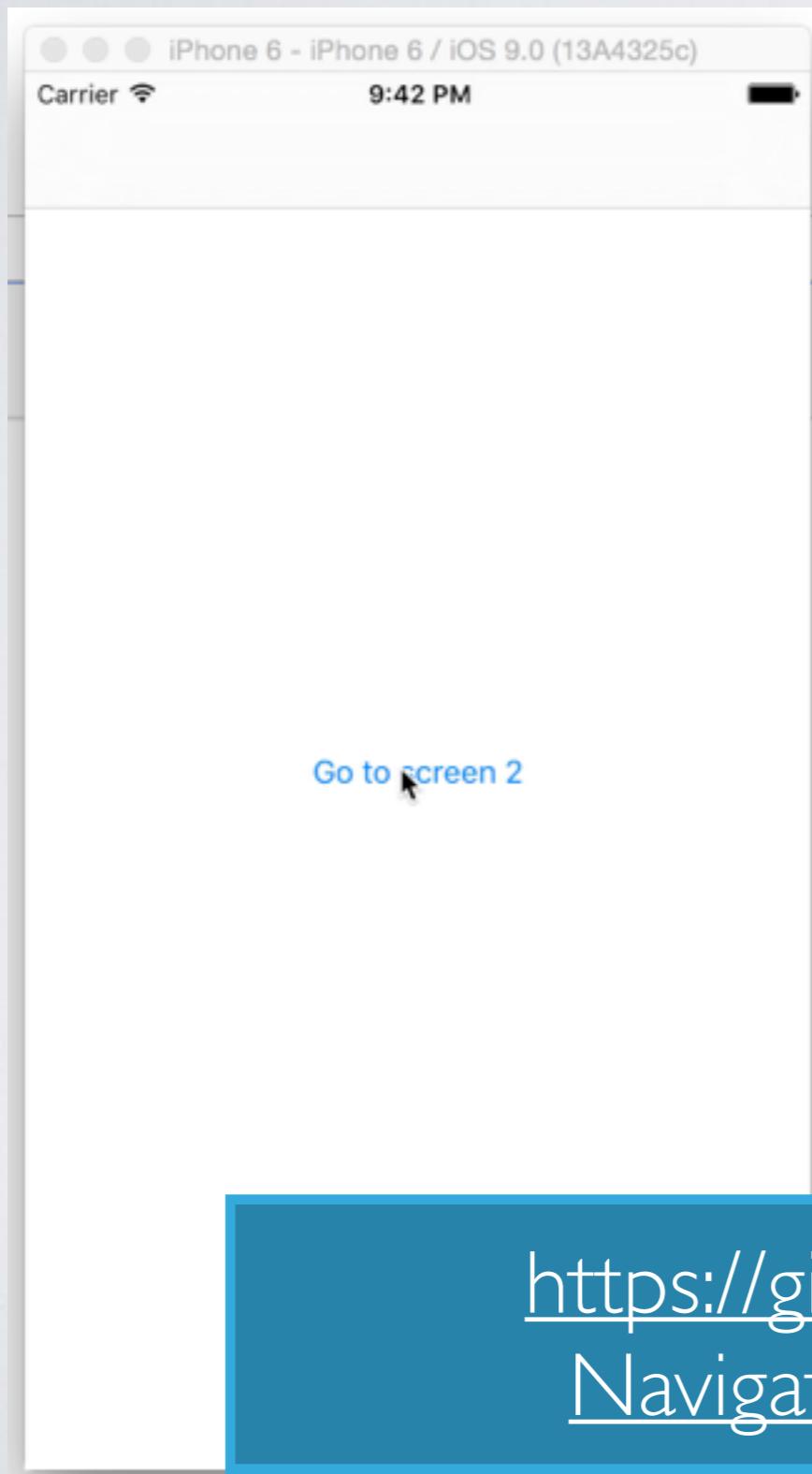
- Underneath, just a stack
- Push a new view
- Pop an old view
- Useful for sequences (drilling down)

NAVIGATIONCONTROLLER



- Underneath, just a stack
- Push a new view
- Pop an old view
- Useful for sequences (drilling down)

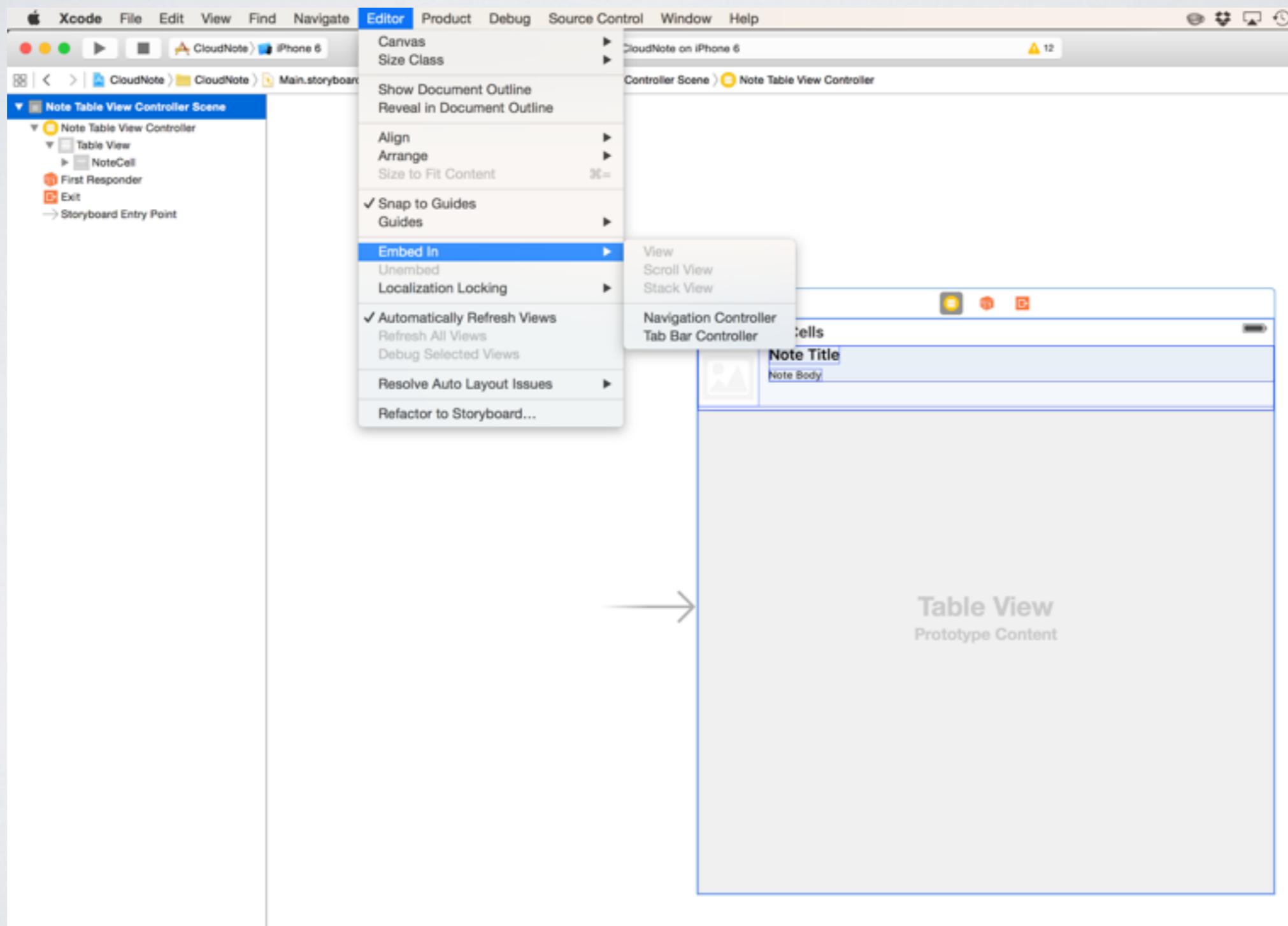
NAVIGATIONCONTROLLER



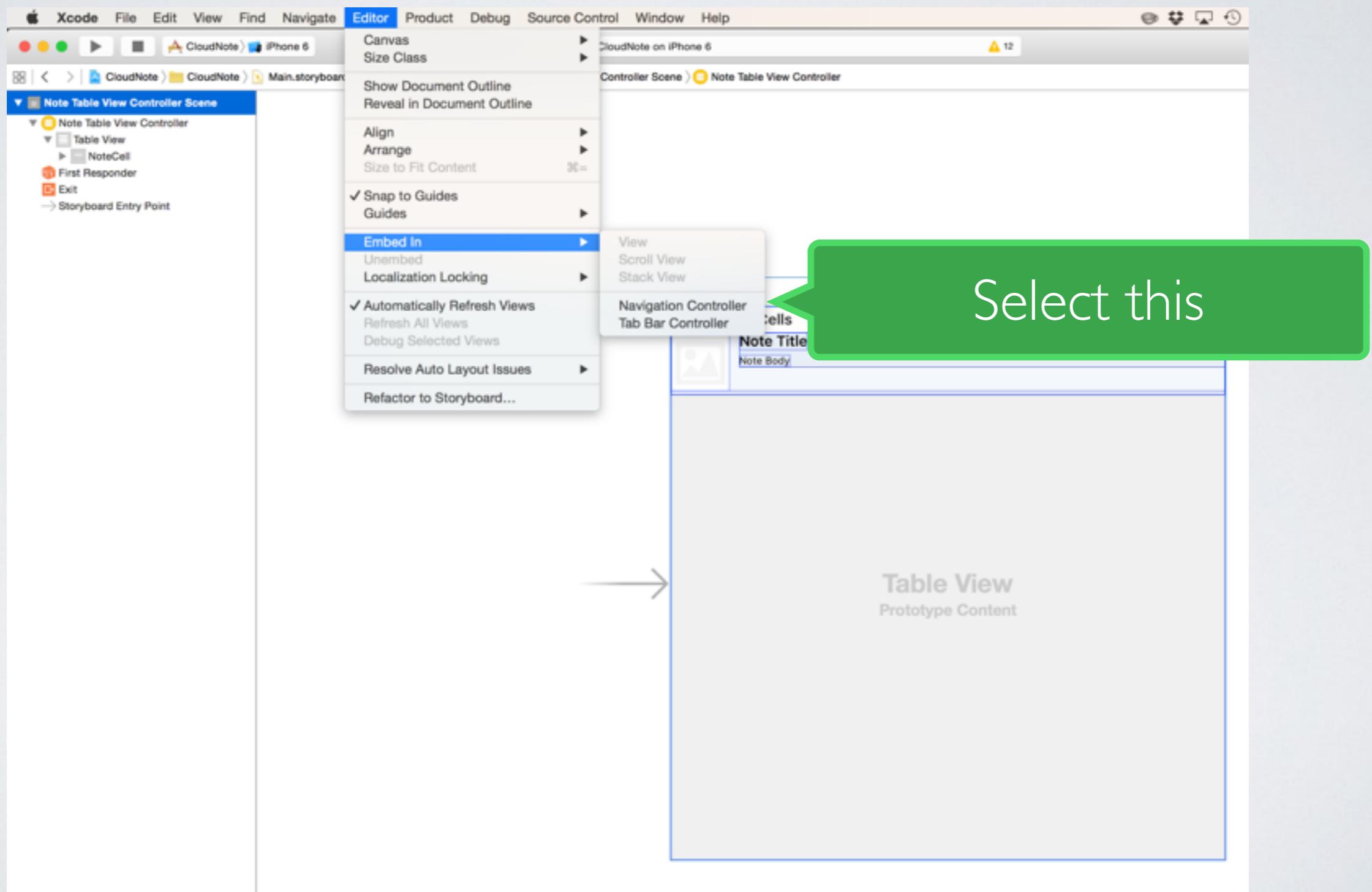
- Underneath, just a stack
- Push a new view
- Pop an old view
- Useful for sequences (drilling down)

[https://github.com/talentsparkio/
NavigationControllerExample](https://github.com/talentsparkio/NavigationControllerExample)

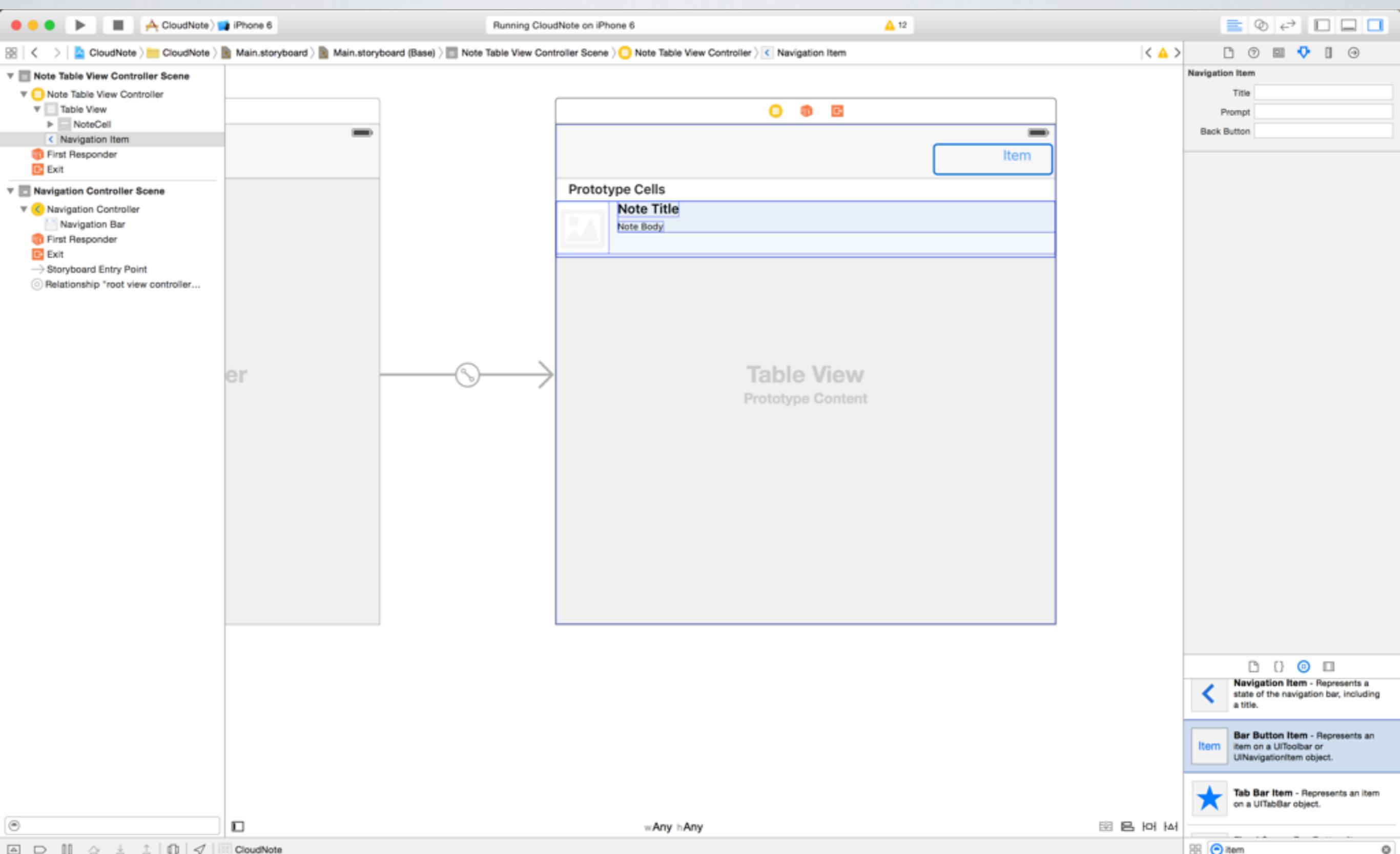
NAVIGATIONCONTROLLER



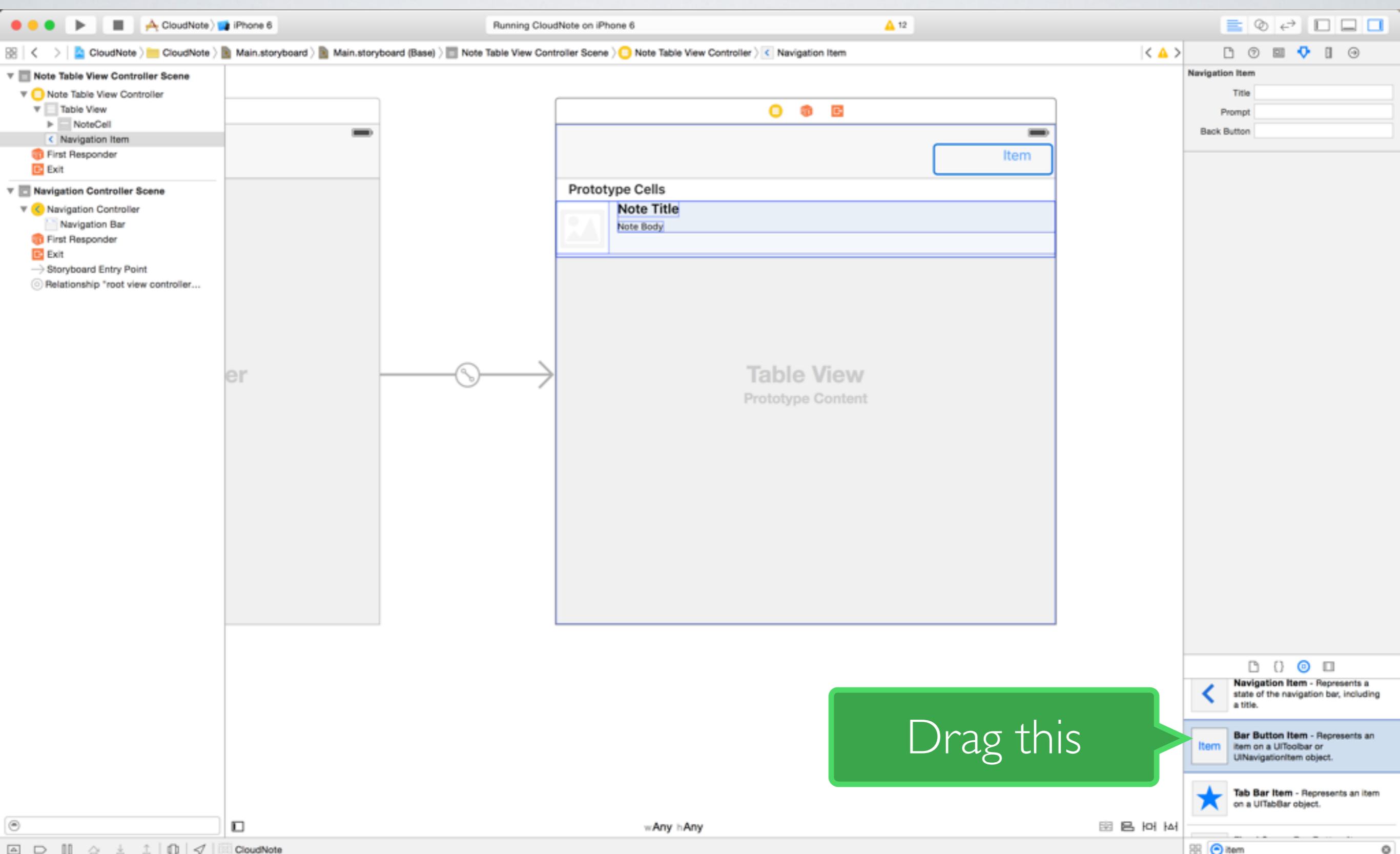
NAVIGATIONCONTROLLER



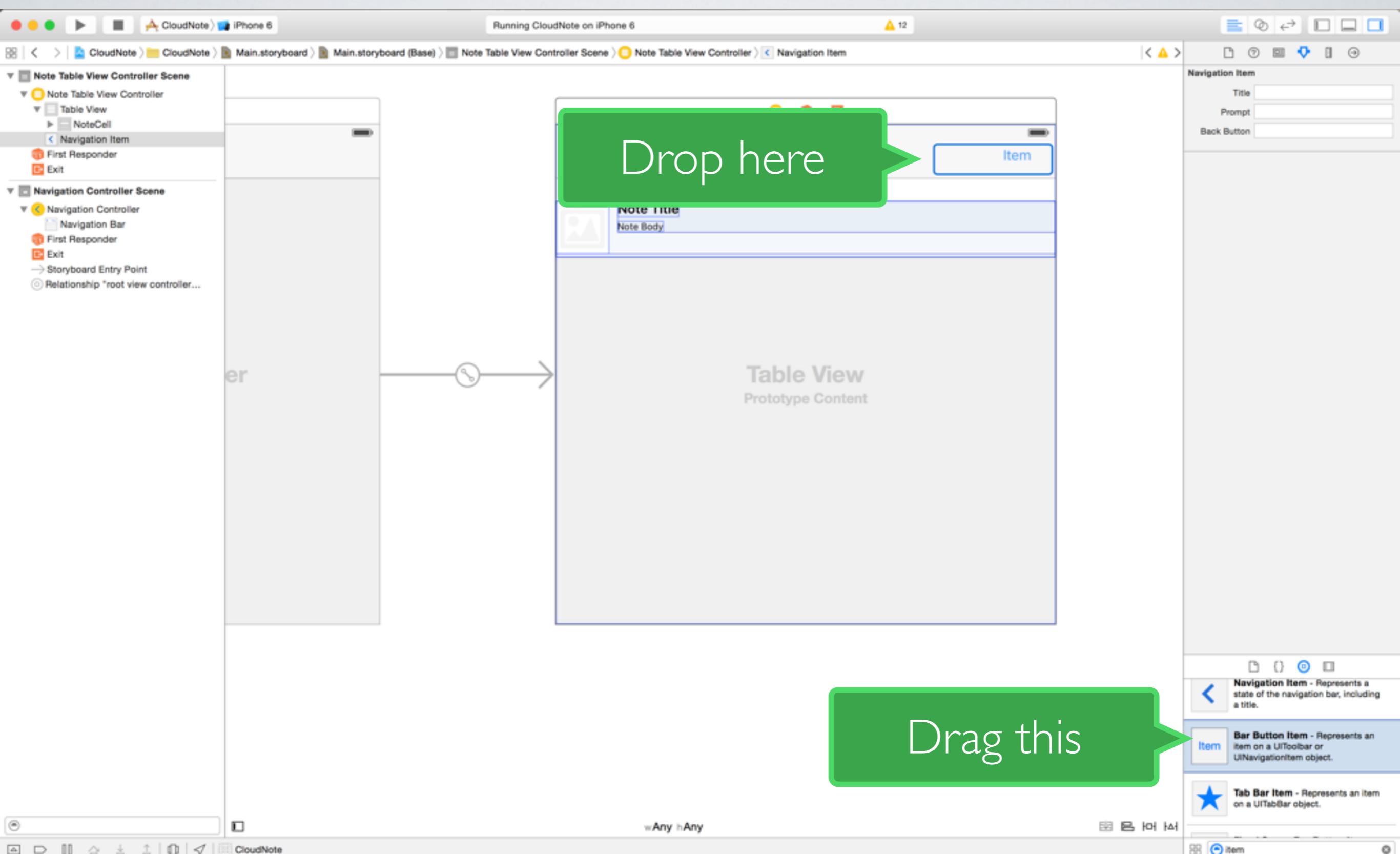
ADD BUTTON



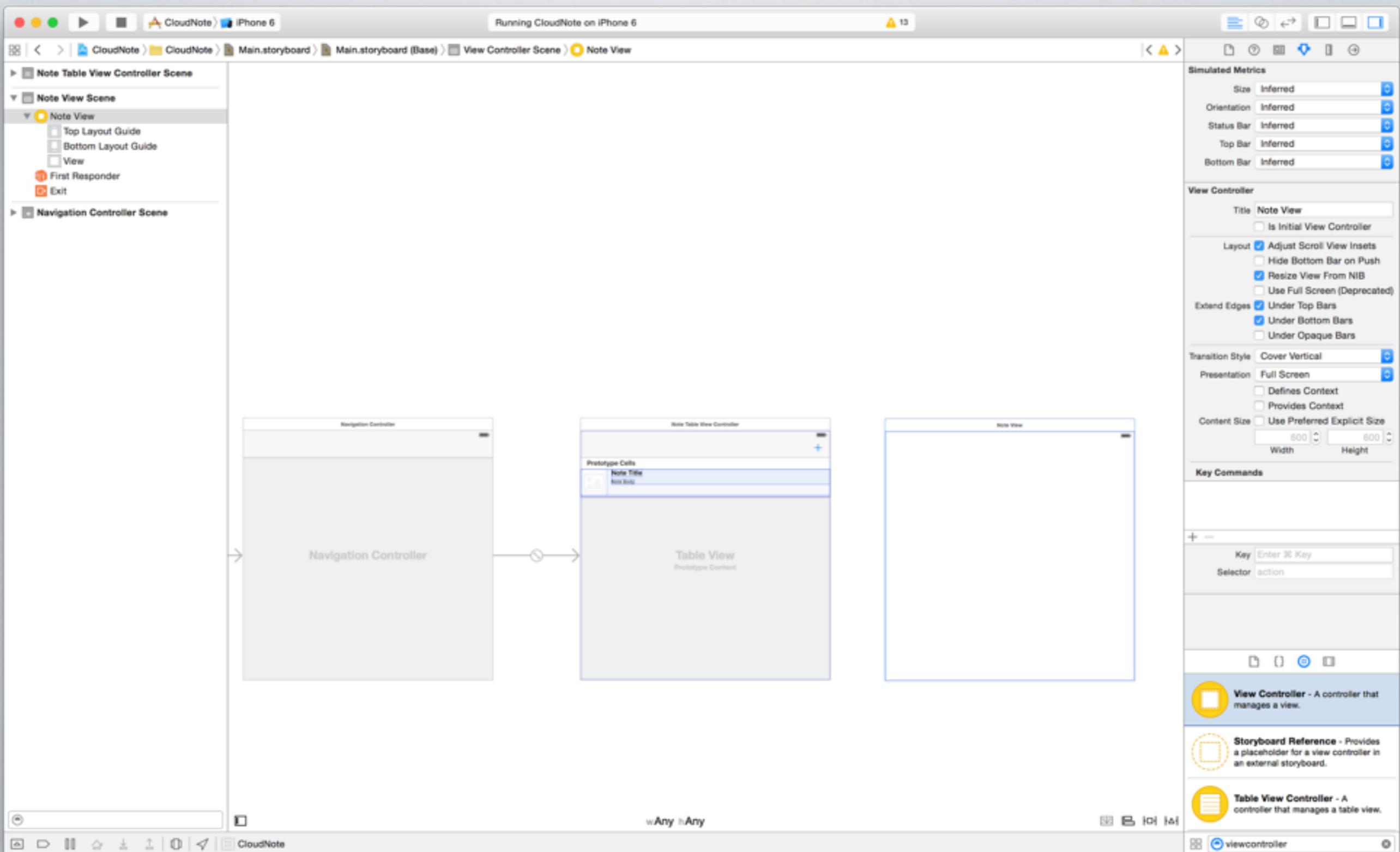
ADD BUTTON



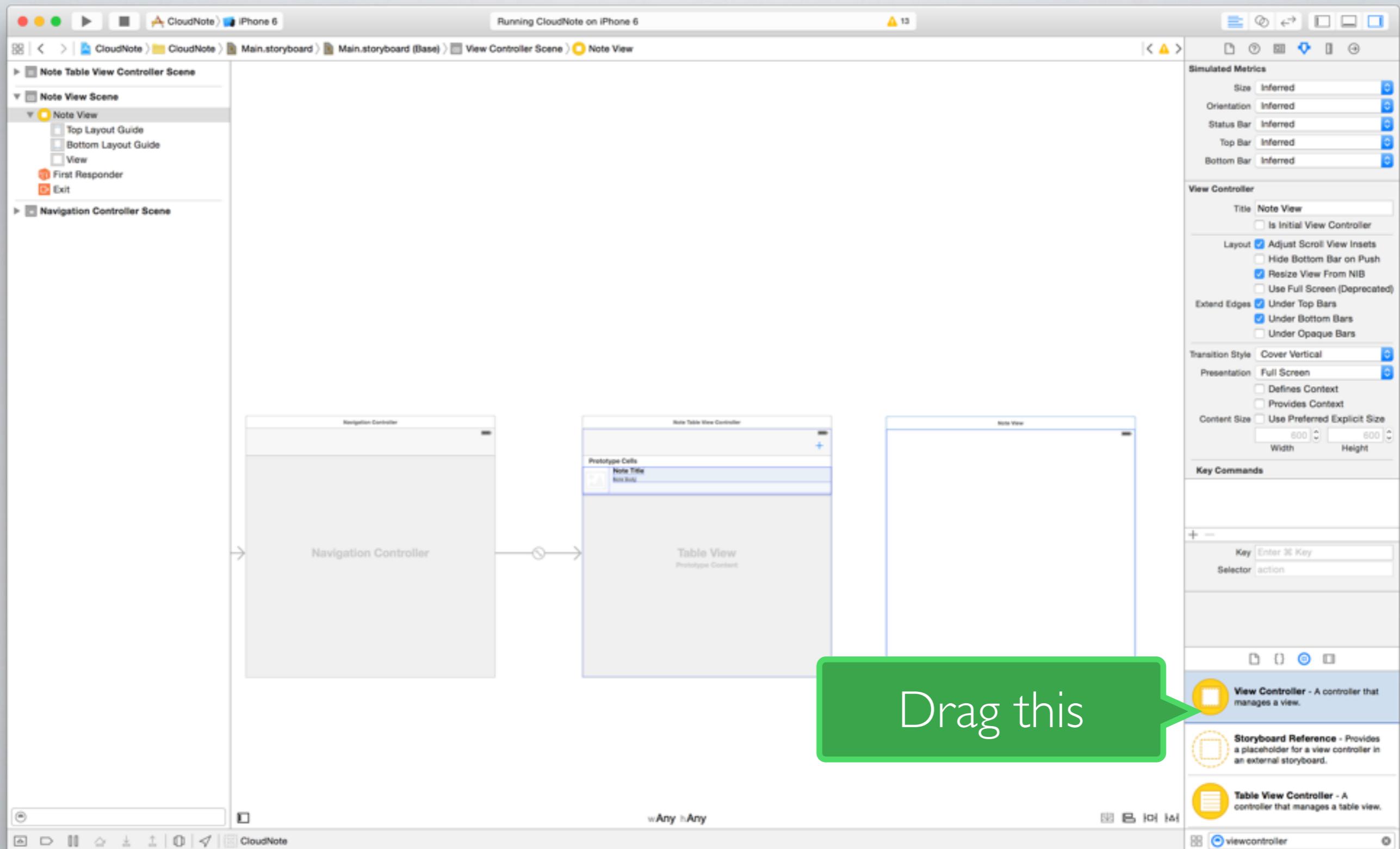
ADD BUTTON



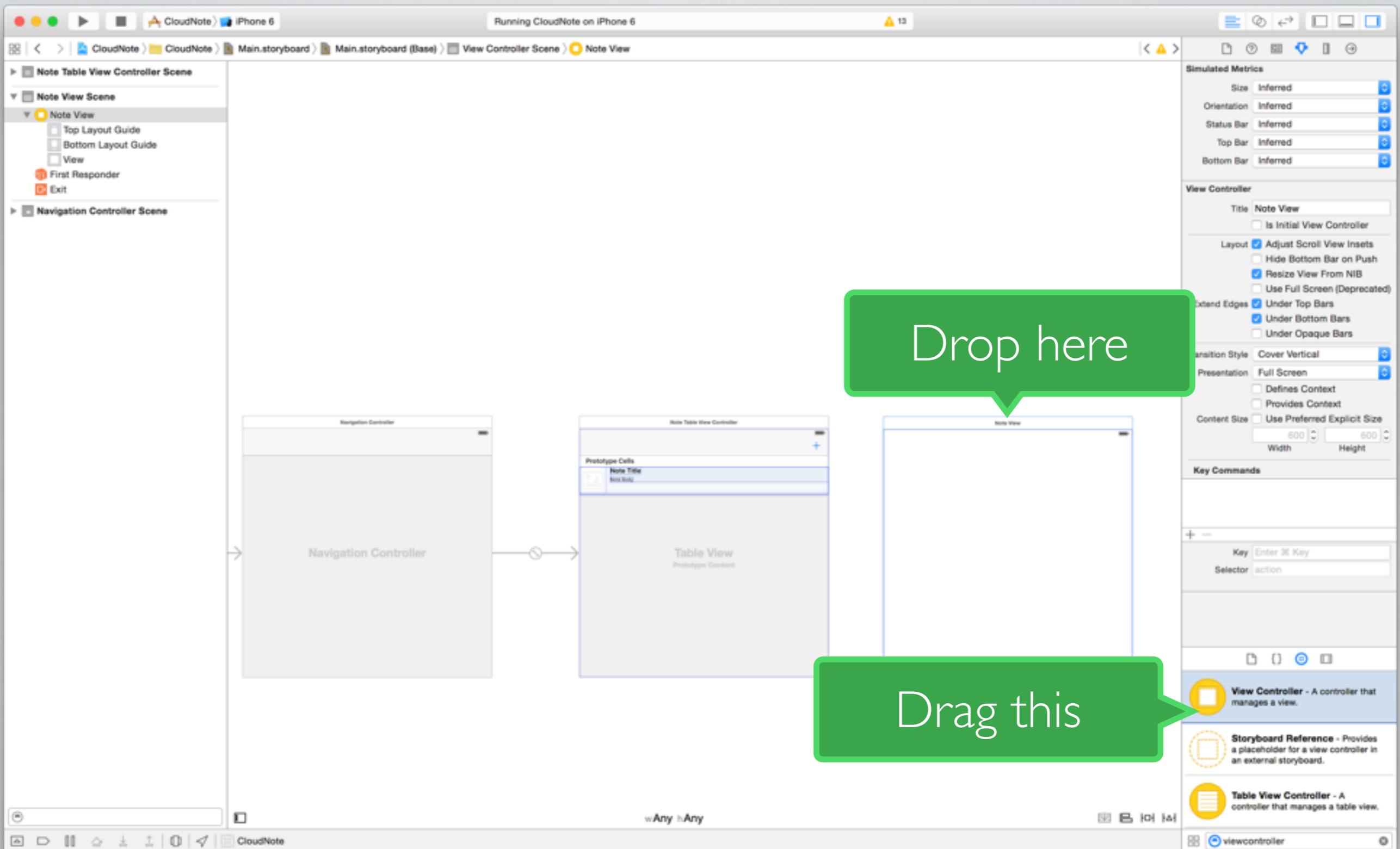
EDIT NOTE VIEW

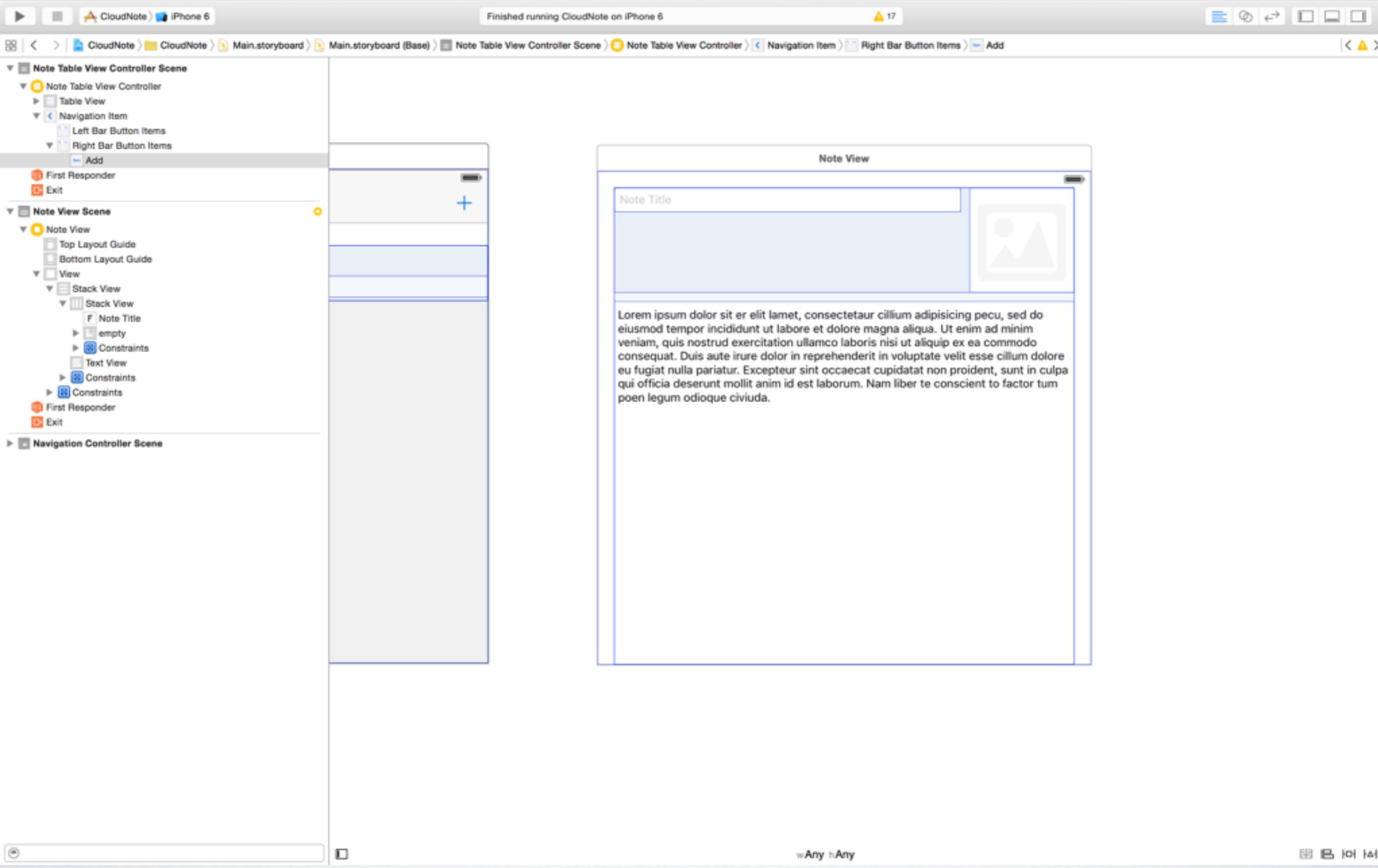


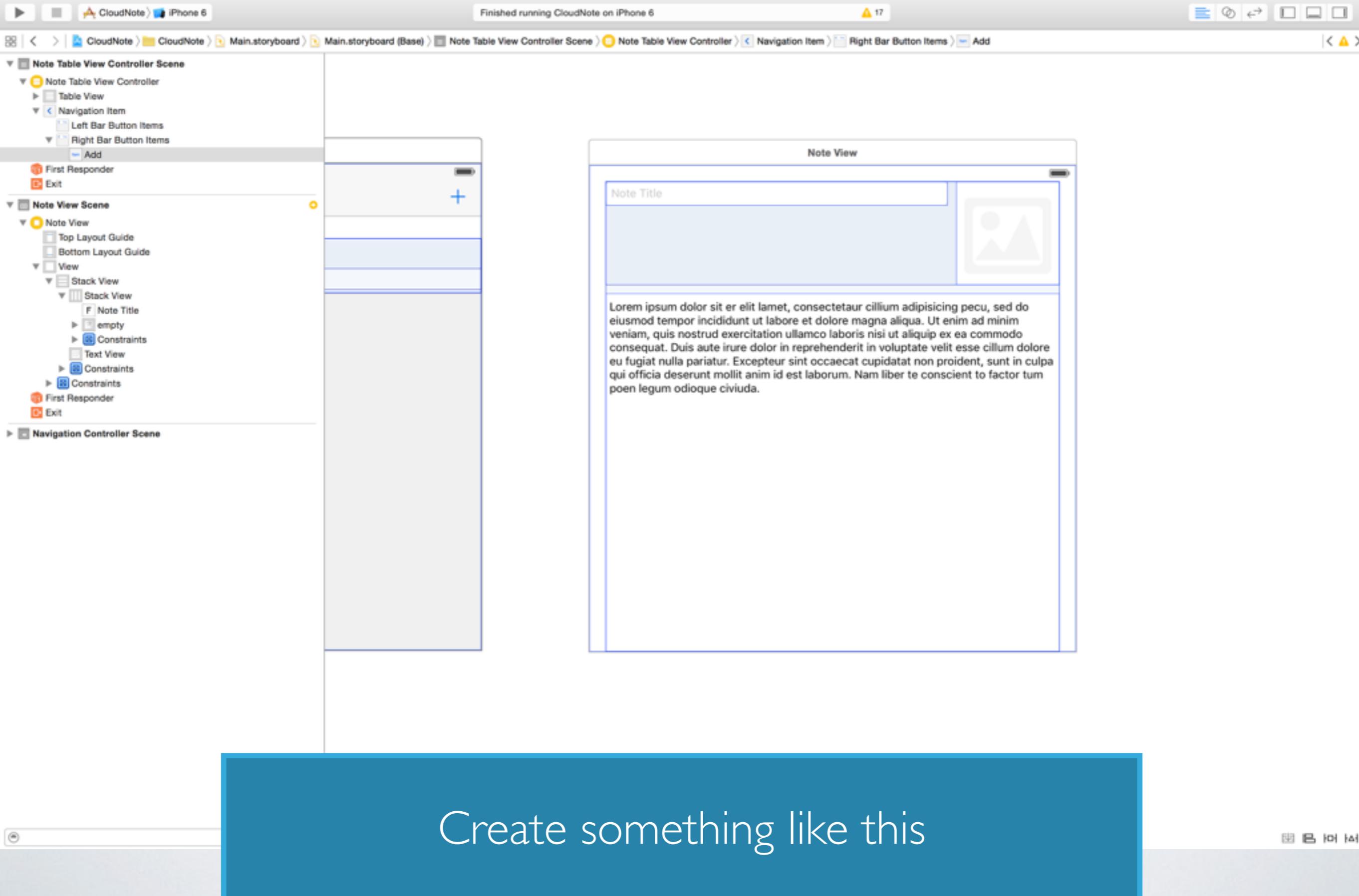
EDIT NOTE VIEW

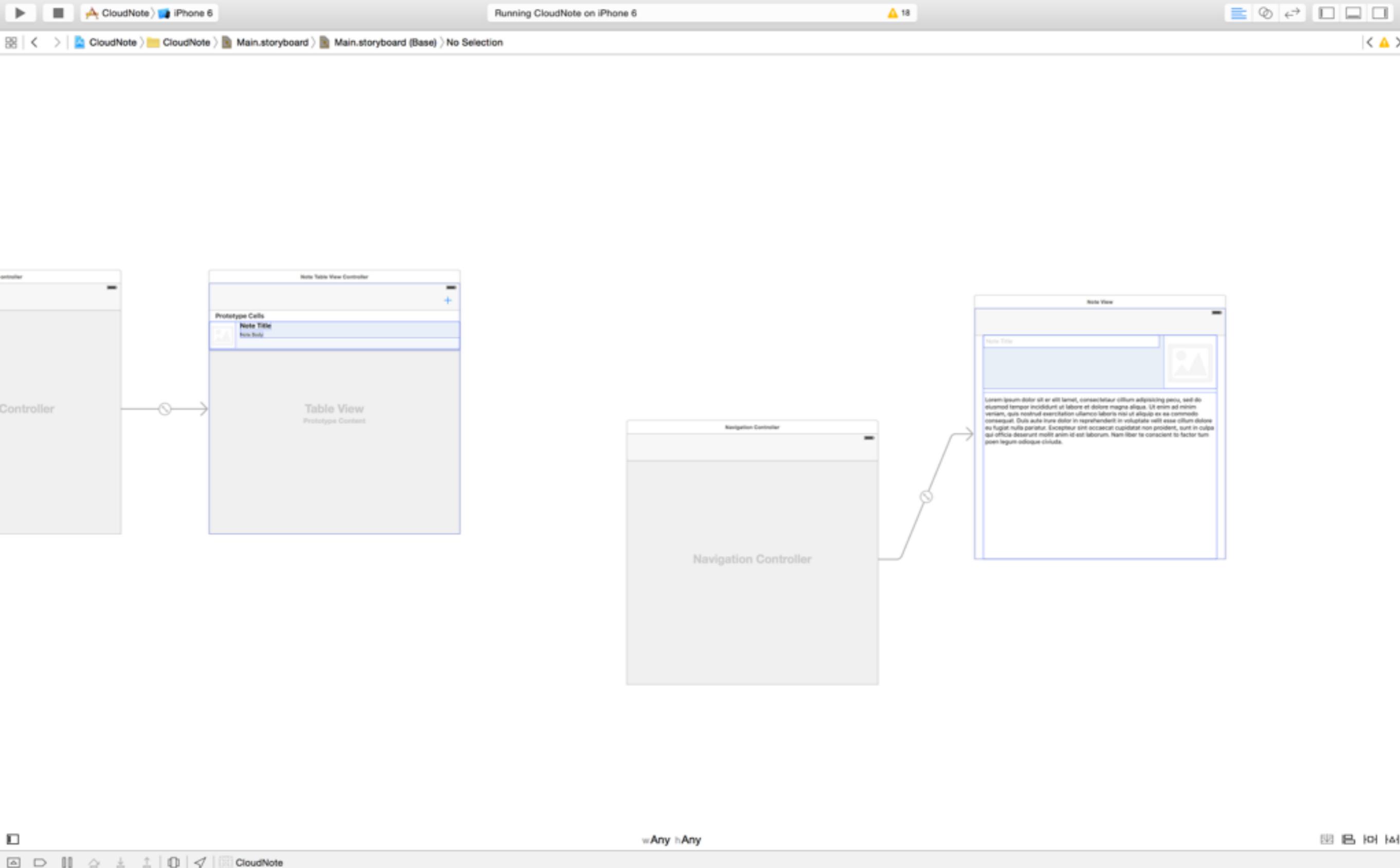


EDIT NOTE VIEW



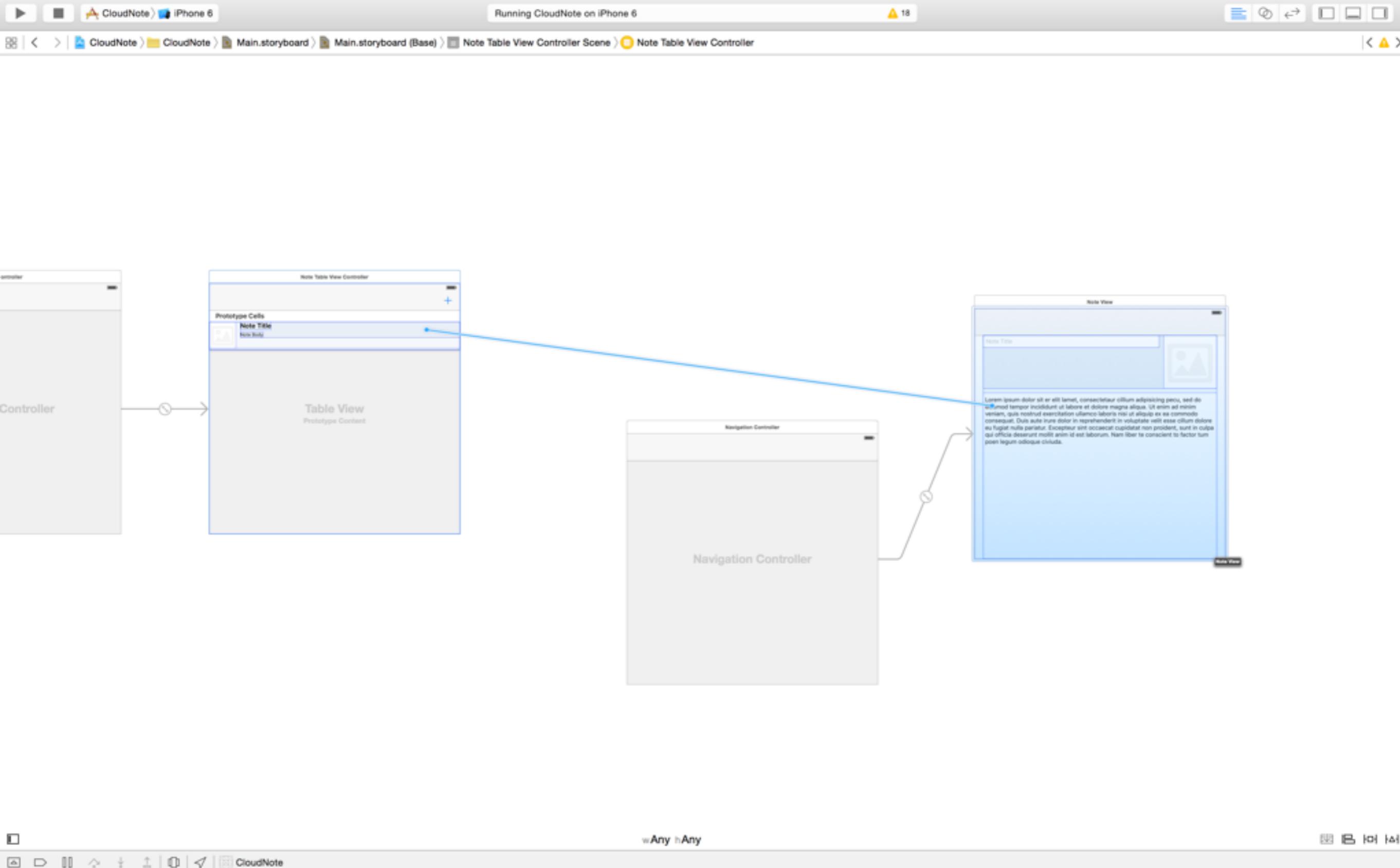


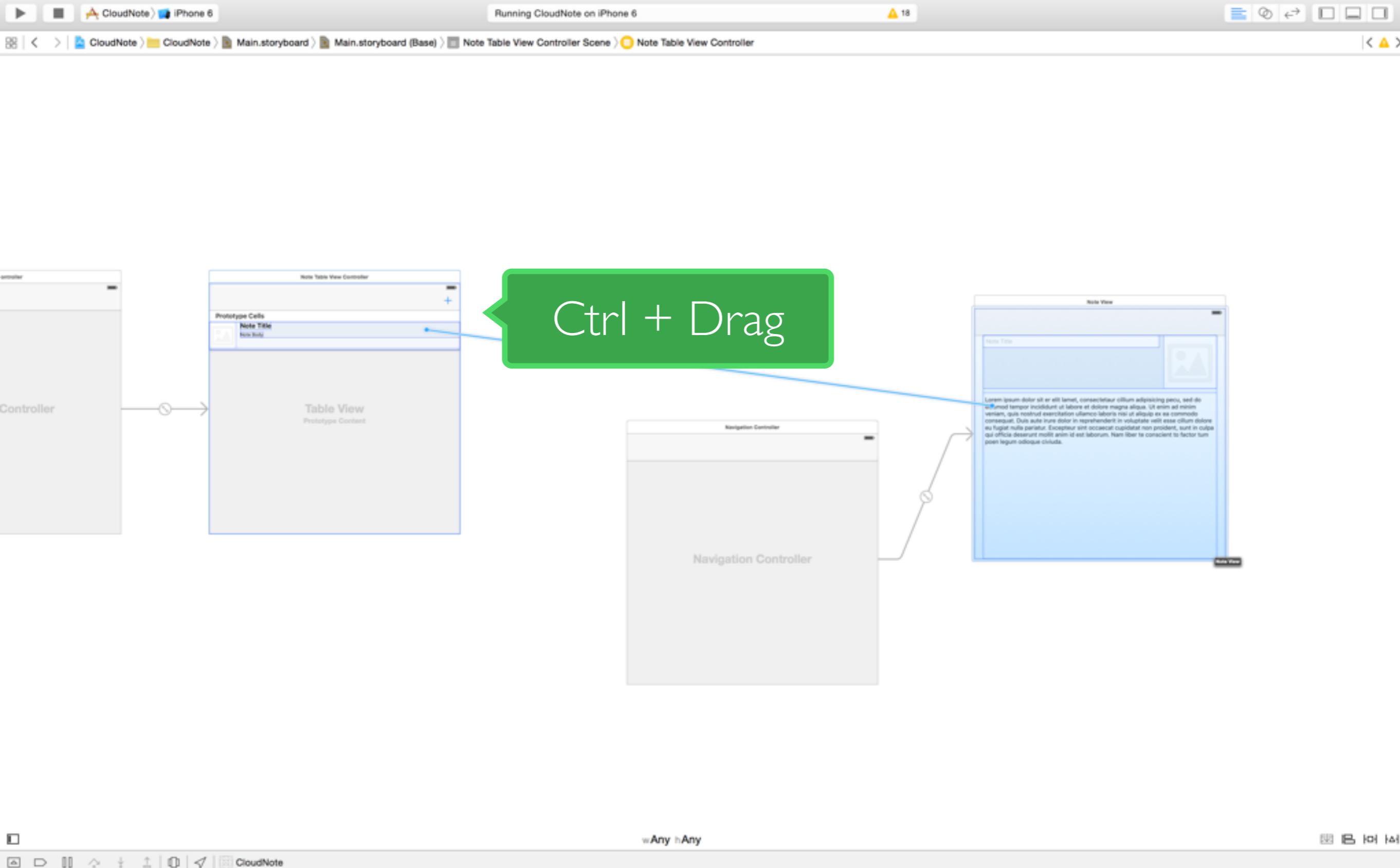


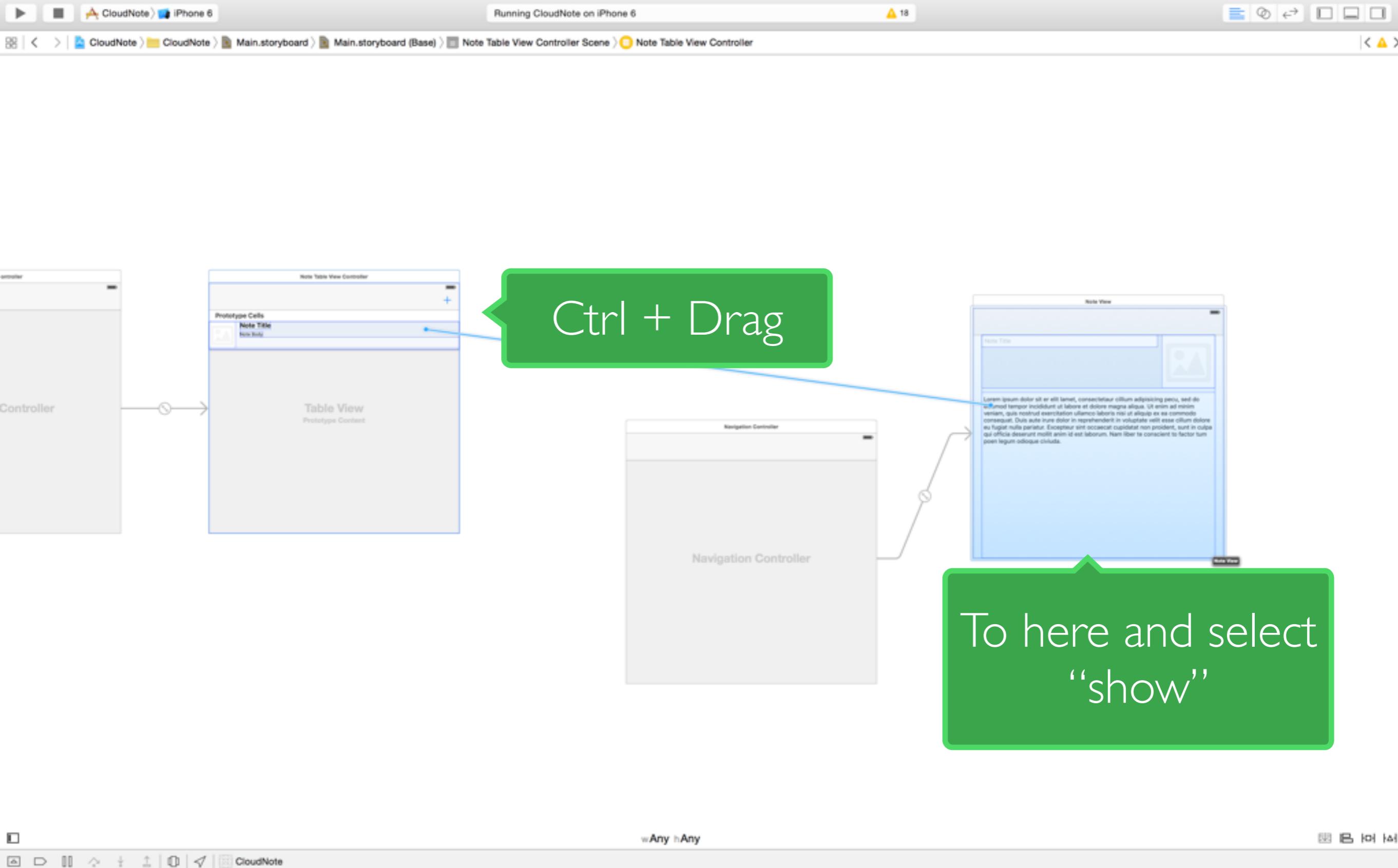


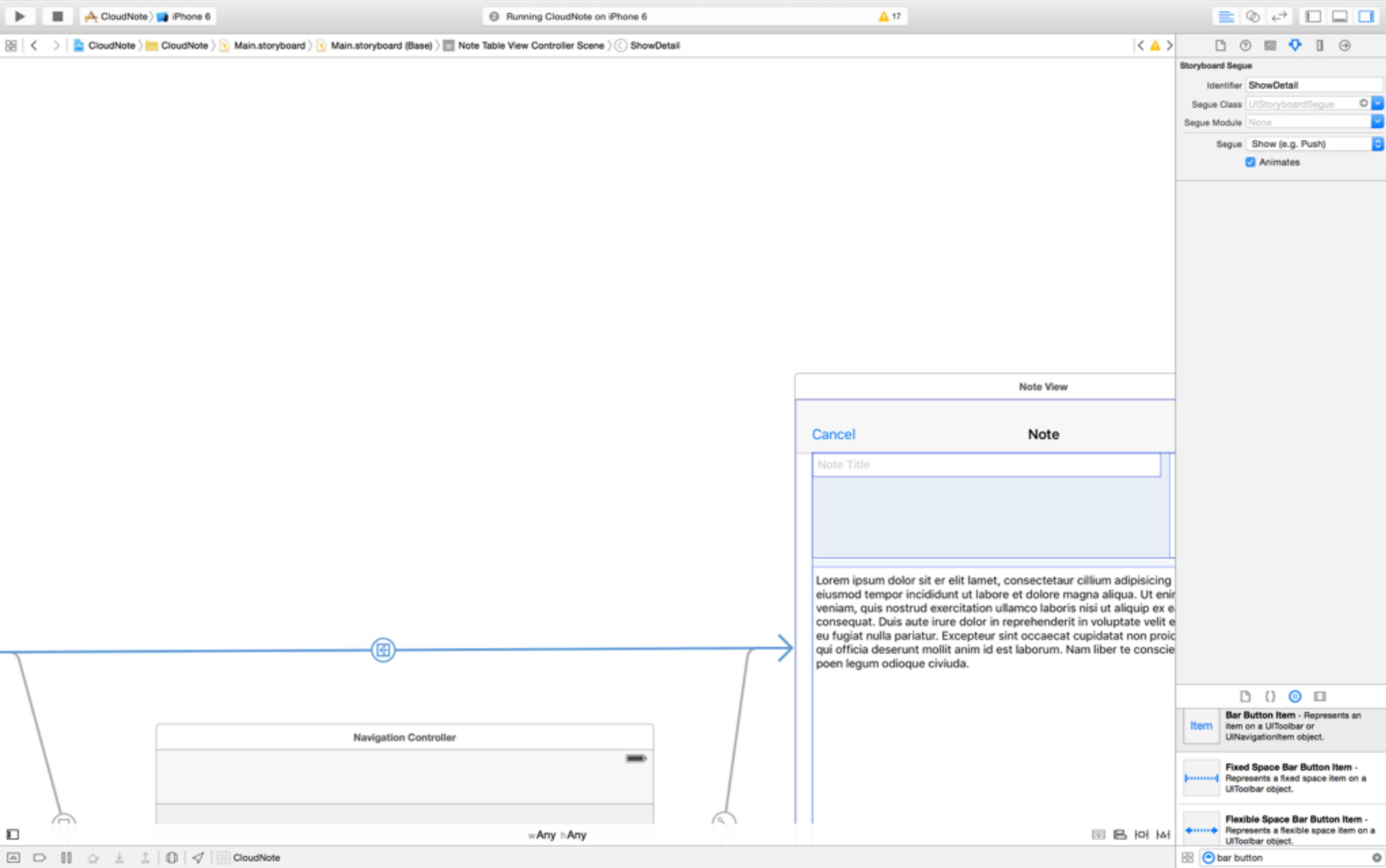


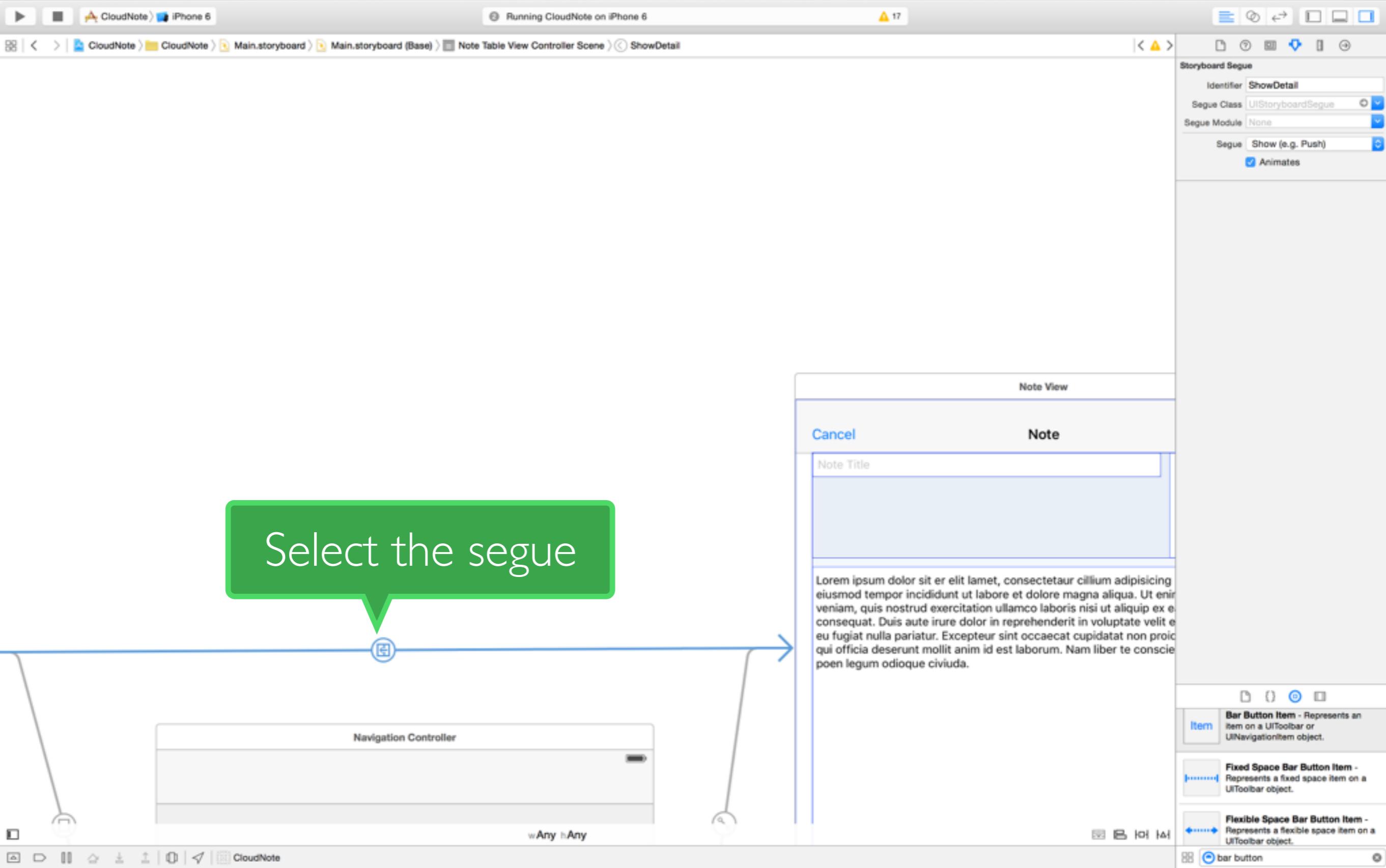
Embed in a UINavigationController as before

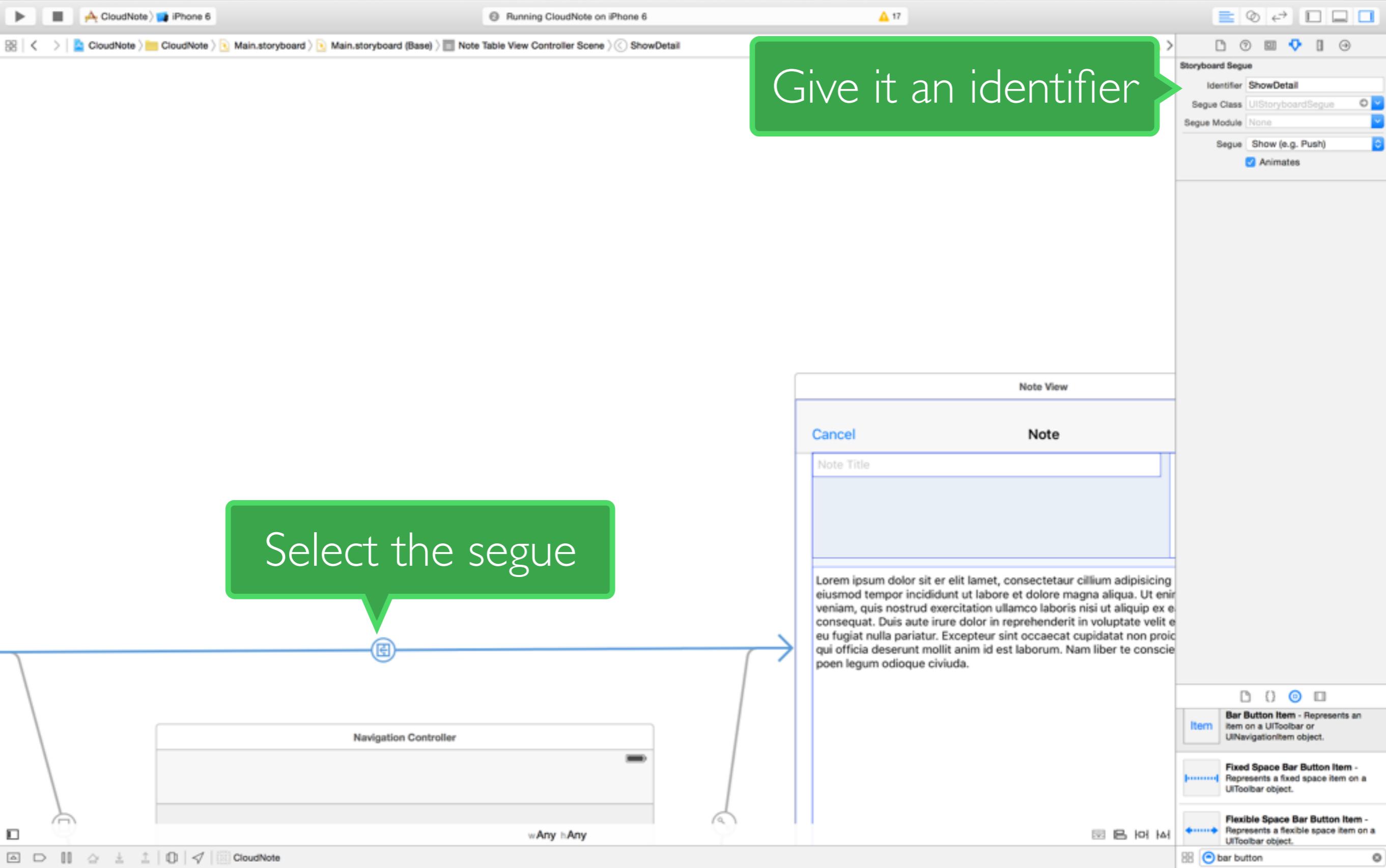


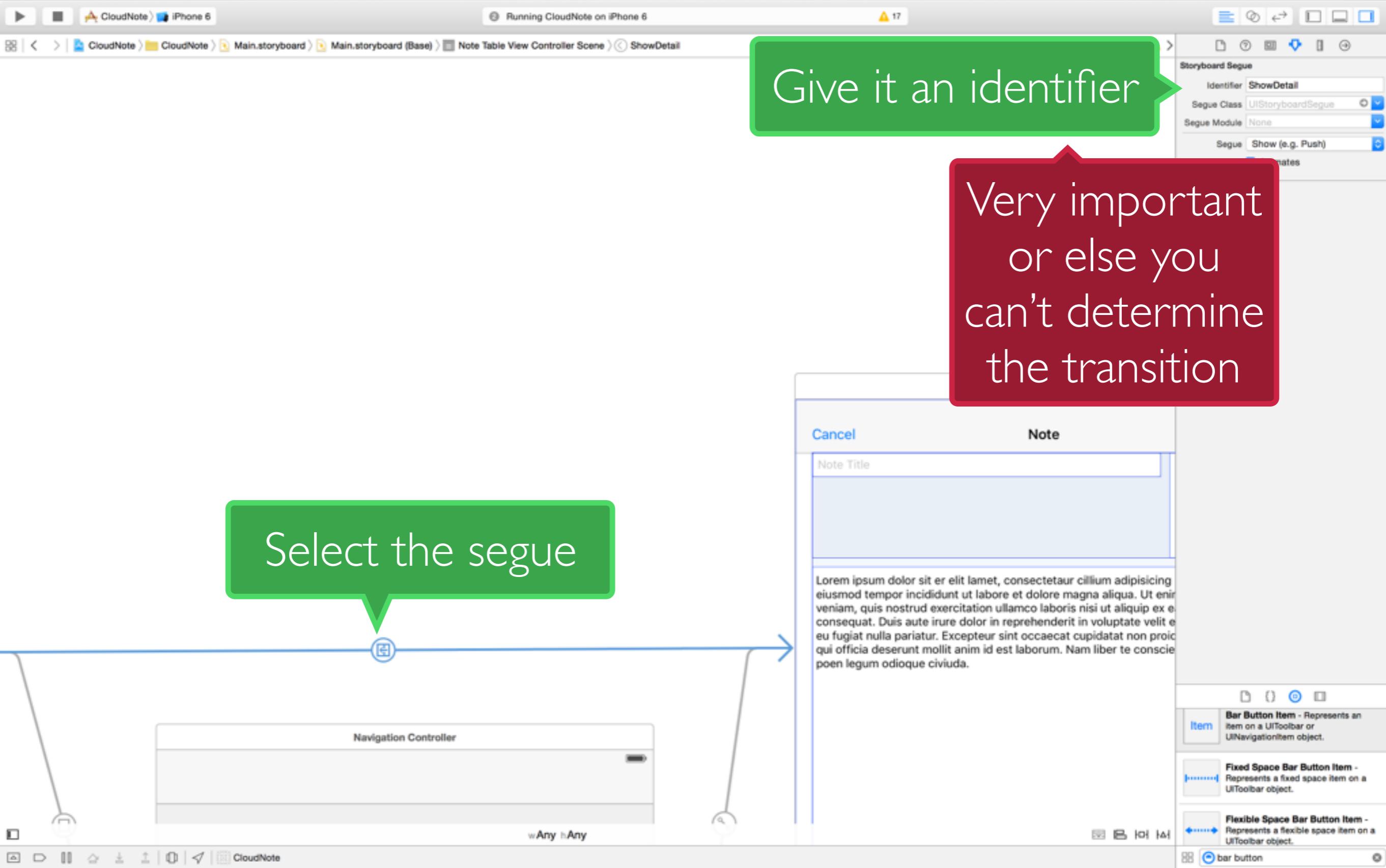


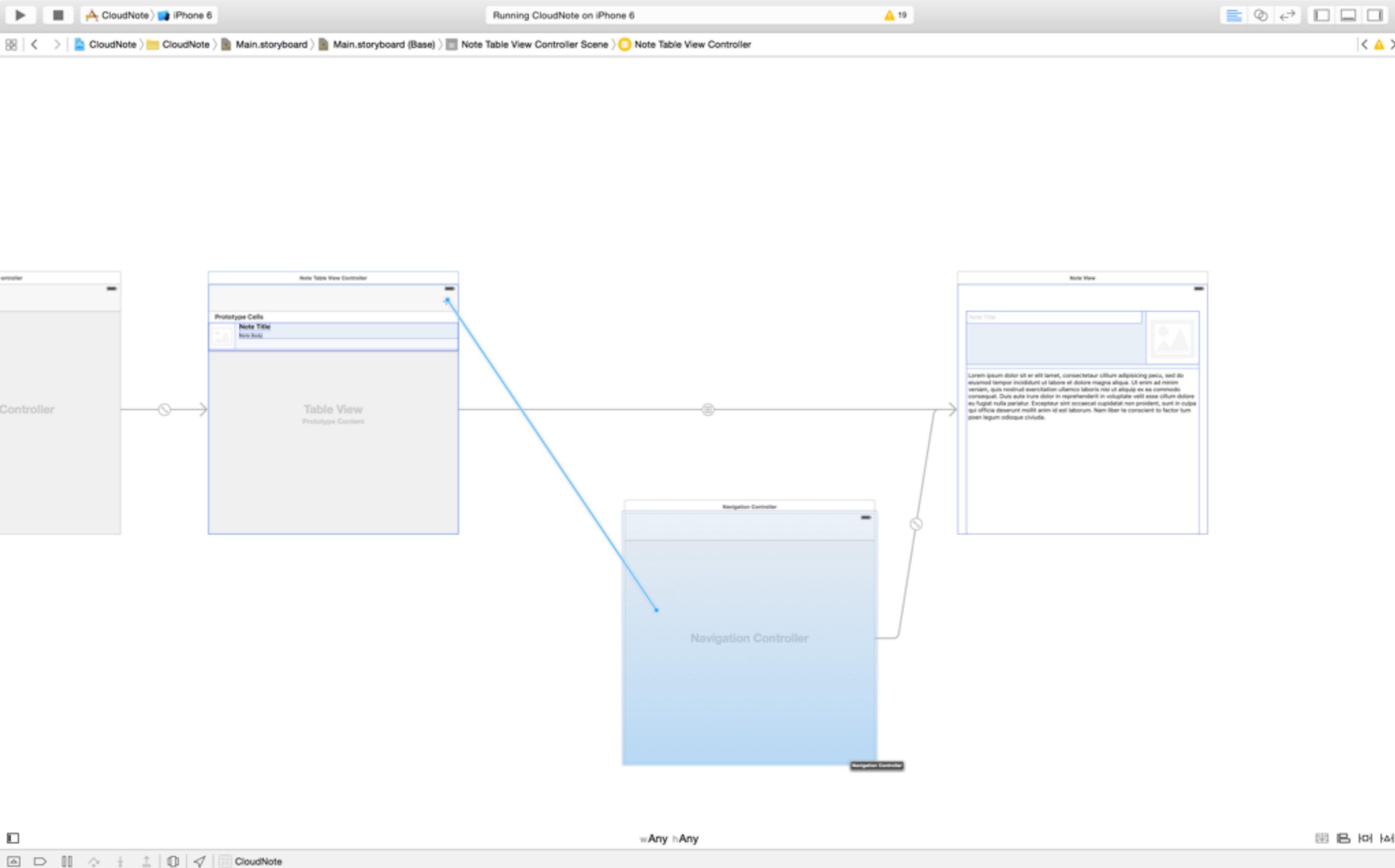


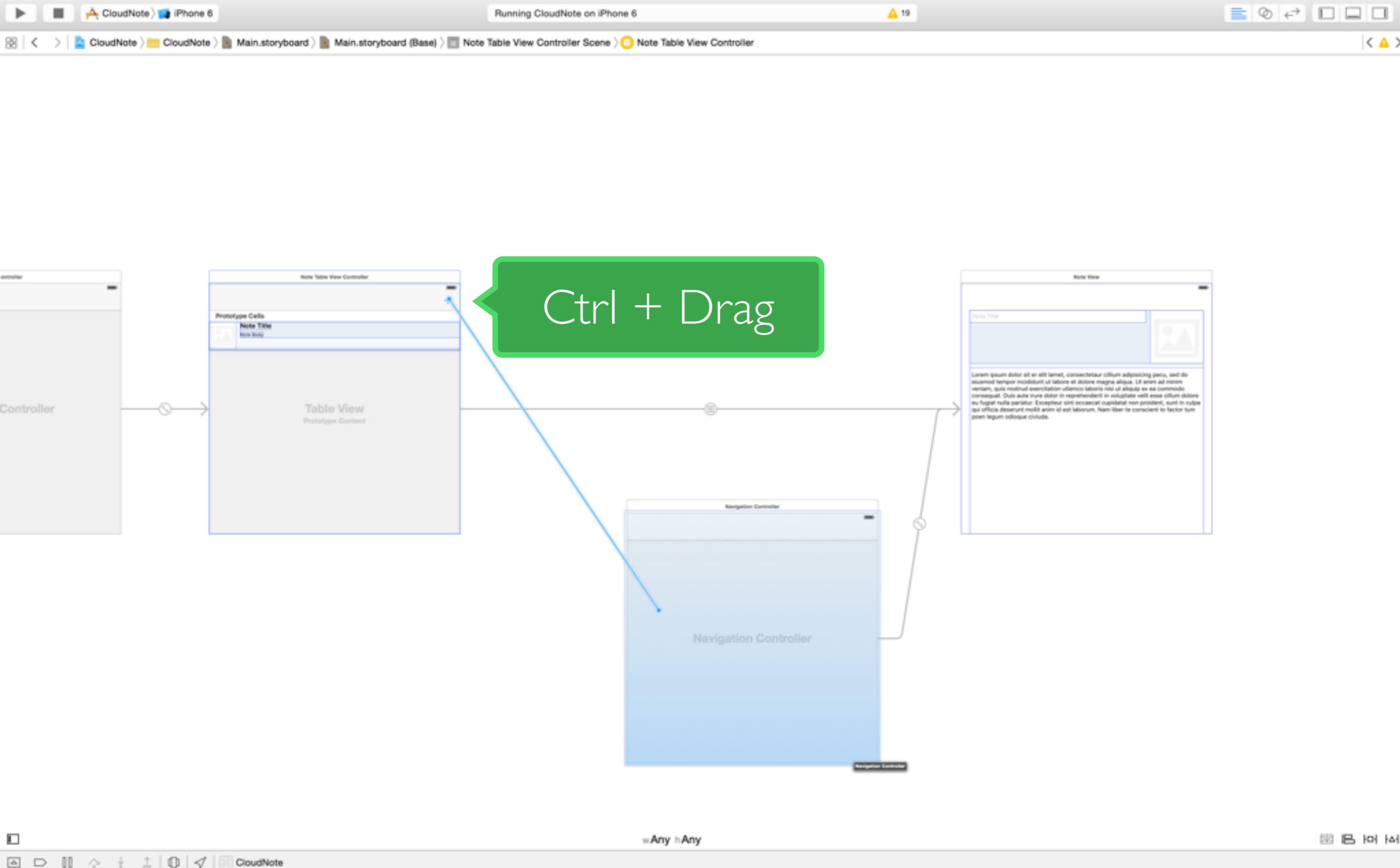


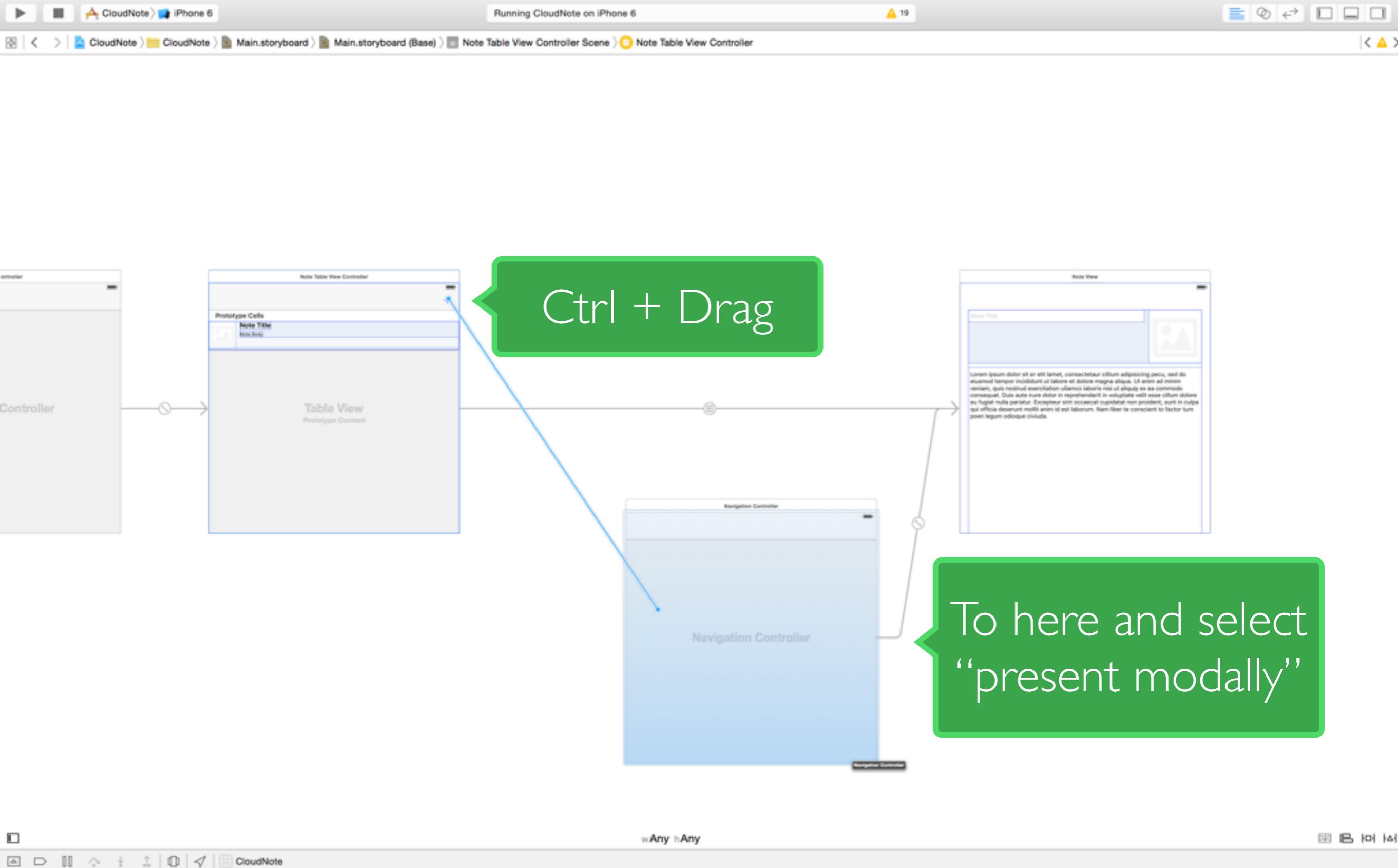


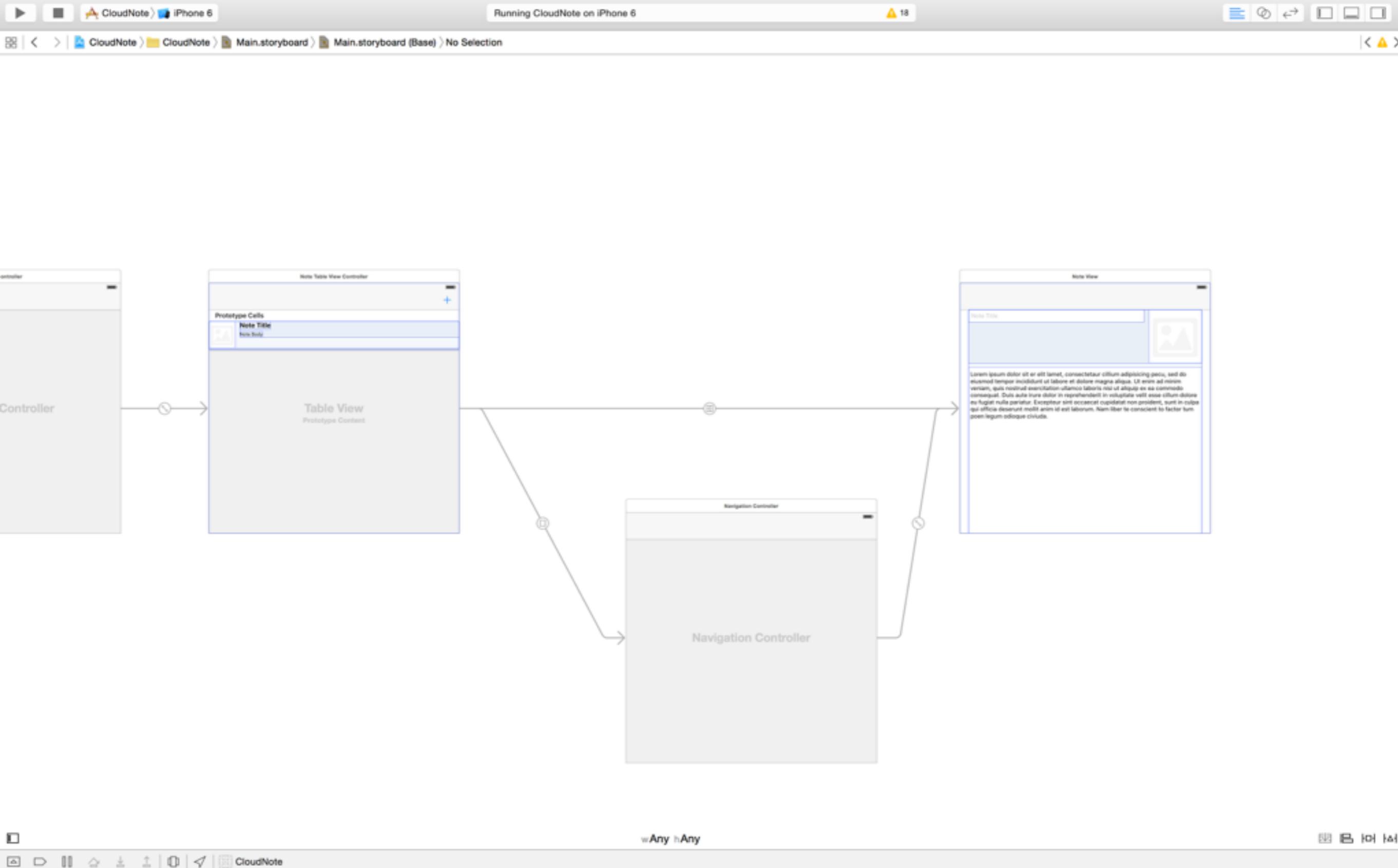


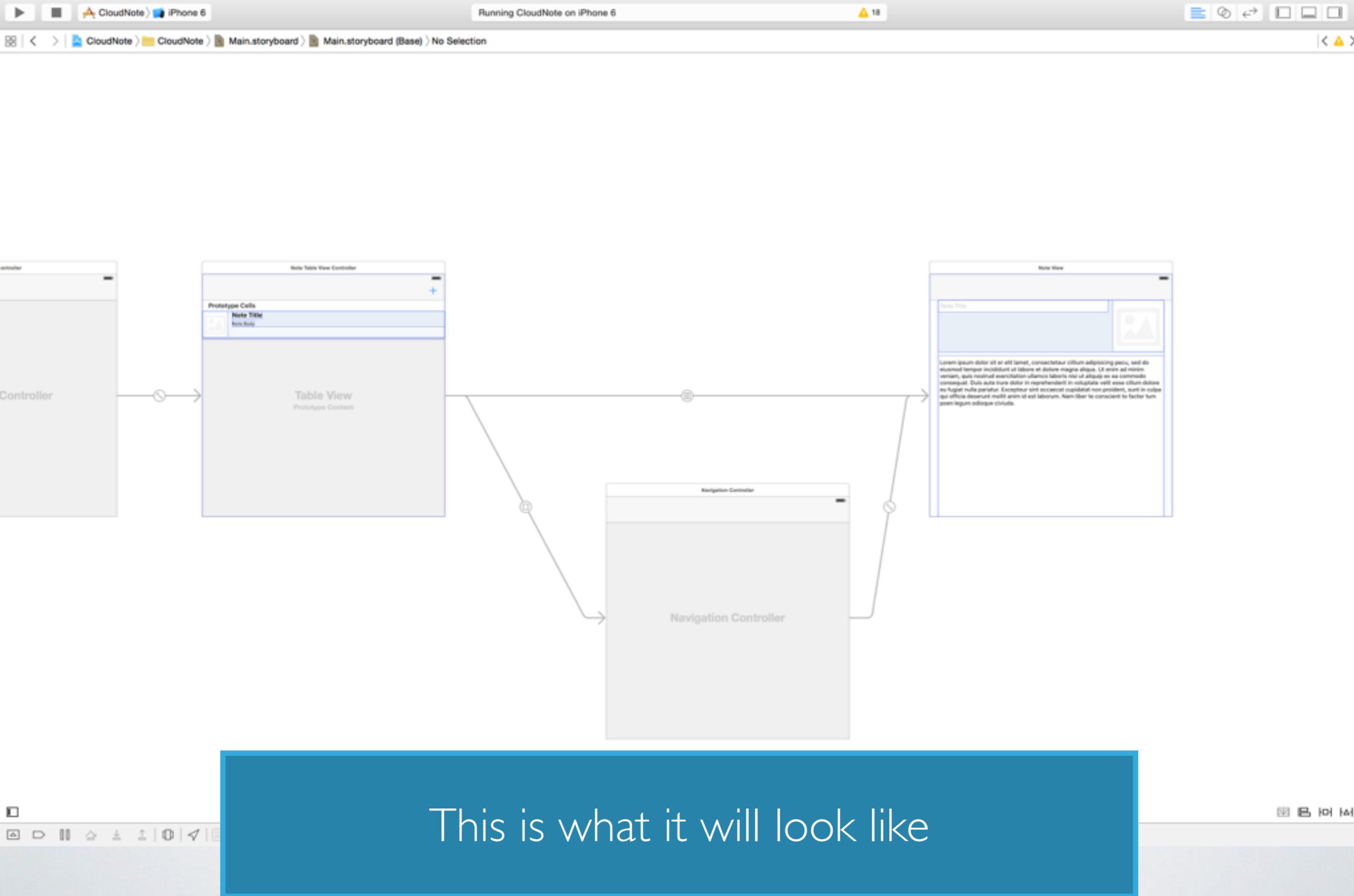






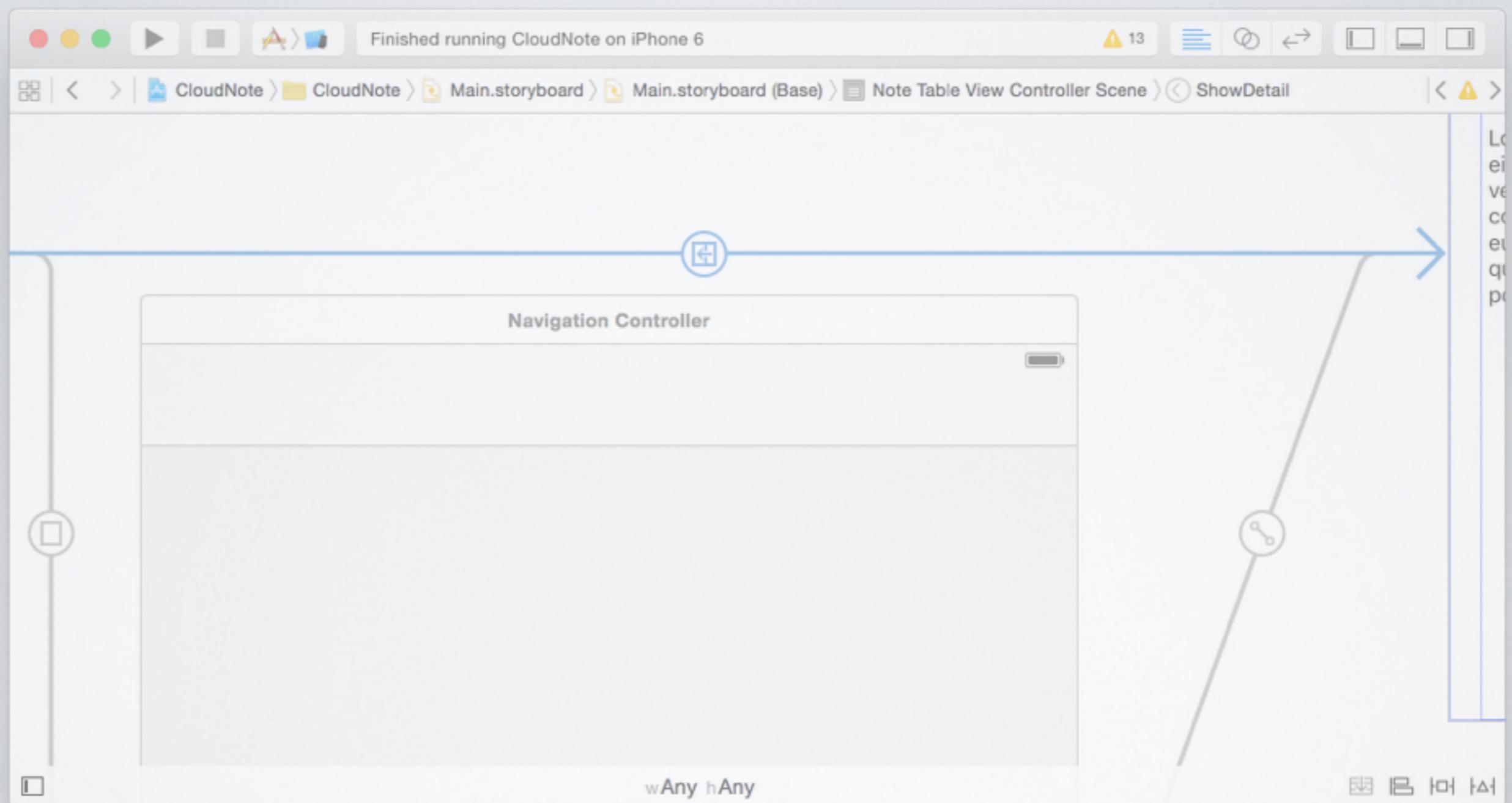




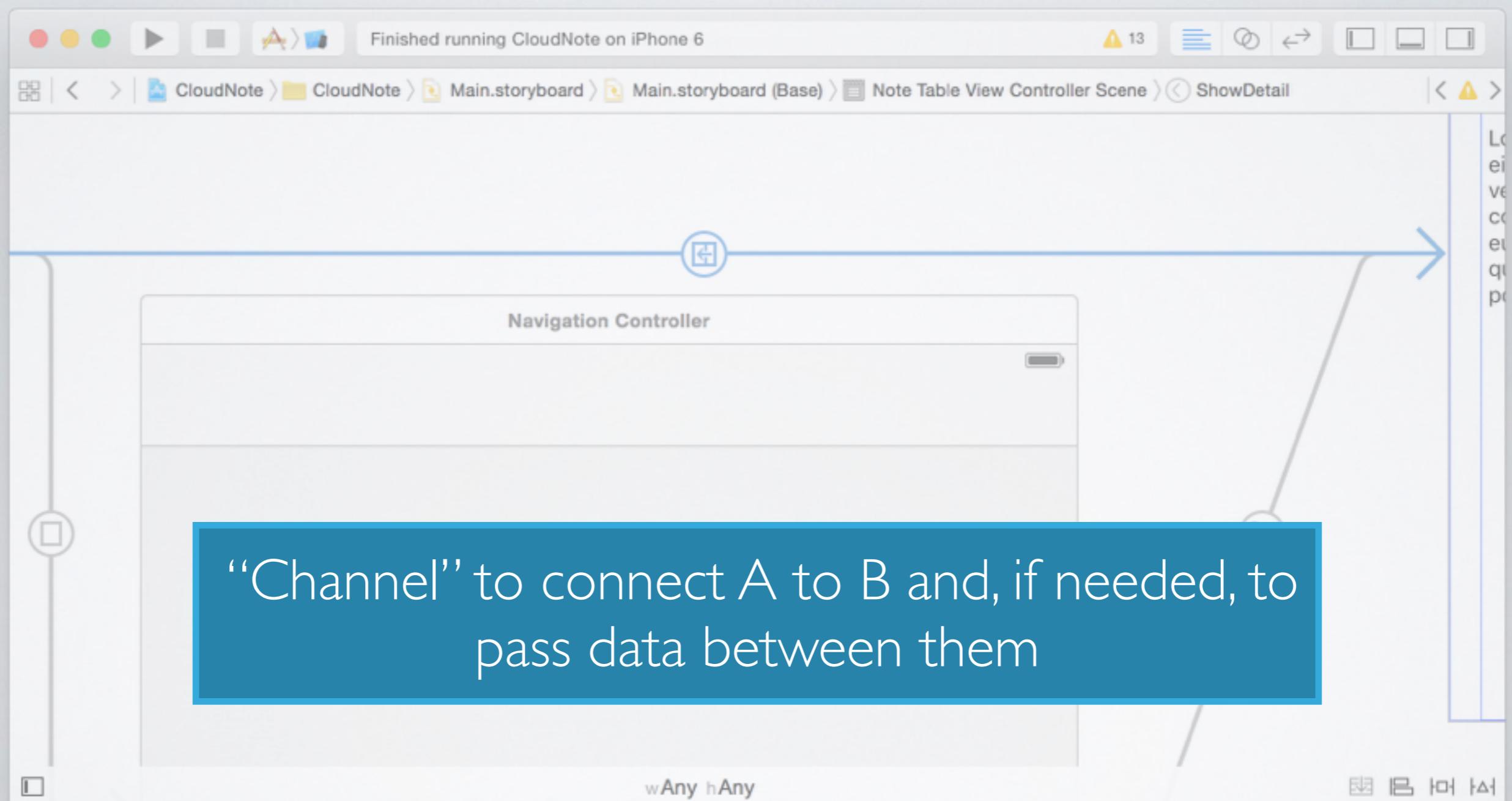


SEGUE

SEGUE



SEGUE



SEGUE

The screenshot shows the Xcode documentation interface. The title bar says "UIStoryboardSegue Class Reference". The search bar contains the text "uistorybooksegue". The main content area displays the class reference for `UIStoryboardSegue`, which is described as a class responsible for performing visual transitions between view controllers. It supports standard UIKit transitions and can be subclassed for custom transitions. The sidebar on the left lists related topics such as "Initializing a Storyboard Segue", "Accessing the Segue Attributes", "Performing the Segue", and "Creating a Custom Segue".

UIStoryboardSegue Class Reference

A `UIStoryboardSegue` object is responsible for performing the visual transition between two view controllers. In addition, segue objects are used to prepare for the transition from one view controller to another. Segue objects contain information about the view controllers involved in a transition. When a segue is triggered, but before the visual transition occurs, the storyboard runtime calls the current view controller's `prepareForSegue:sender:` method so that it can pass any needed data to the view controller that is about to be displayed.

The `UIStoryboardSegue` class supports the standard visual transitions available in UIKit. You can also subclass to define custom transitions between the view controllers in your storyboard file.

You do not create segue objects directly. Instead, the storyboard runtime creates them when it must perform a segue between two view controllers. You can still initiate a segue programmatically using the `performSegueWithIdentifier:sender:` method of `UIViewController` if you want. You might do so to initiate a segue from a source that was added programmatically and therefore not available in Interface Builder.

Subclassing Notes

- Initialization:
 - `init(identifier:source:destination:)`
- Accessing attributes:
 - `sourceViewController`
 - `destinationViewController`
 - `identifier`
- Performing:
 - `perform()`
- Creating a Custom Segue

SEGUE

The screenshot shows the Xcode documentation browser with the search bar containing "uistorybooksegue". The results list shows "UIStoryboardSegue" as the top result, followed by "UIStoryboardPopoverSegue", "Google", and "Stack Overflow". The main content area displays the "UIStoryboardSegue Class Reference". It includes a brief description of the class's purpose, a sidebar with sections like "Initializing a Storyboard Segue" and "Accessing the Segue Attributes", and a green callout bubble pointing to the "Where from?" section.

UIStoryboardSegue Class Reference

A UIStoryboardSegue object is responsible for performing the visual transition between two view controllers. In addition, segue objects are used to prepare for the transition from one view controller to another. Segue objects contain information about the view controllers involved in a transition. When a segue is triggered, but before the visual transition occurs, the storyboard runtime calls the current view controller's `prepareForSegue:sender:` method so that it can pass any needed data to the view controller that is about to be displayed.

The UIStoryboardSegue class supports the standard visual transitions available in UIKit. You can also subclass to define custom transitions between the view controllers instead, the storyboard runtime creates them to view controllers. You can still initiate a `segueWithIdentifier:sender:` method of do so to initiate a segue from a source that was added programmatically and therefore not available in Interface Builder.

Where from?

Subclassing Notes

Initializing a Storyboard Segue

M `init(identifier:source:destination:)`

Accessing the Segue Attributes

P `sourceViewController`

P `destinationViewController`

P `identifier`

Performing the Segue

M `perform()`

Creating a Custom Segue

M `init(identifier:source:destination:)`

SEGUE

The screenshot shows the Xcode documentation interface. The title bar says "UIStoryboardSegue Class Reference". The search bar contains the text "uistorybooksegue". The main content area displays the class reference for `UIStoryboardSegue`, which is described as a class responsible for performing visual transitions between view controllers. It supports standard UIKit transitions and can be subclassed for custom transitions. The sidebar on the left lists related topics such as "Initializing a Storyboard Segue", "Accessing the Segue Attributes", "Performing the Segue", and "Creating a Custom Segue".

UIStoryboardSegue Class Reference

A `UIStoryboardSegue` object is responsible for performing the visual transition between two view controllers. In addition, segue objects are used to prepare for the transition from one view controller to another. Segue objects contain information about the view controllers involved in a transition. When a segue is triggered, but before the visual transition occurs, the storyboard runtime calls the current view controller's `prepareForSegue:sender:` method so that it can pass any needed data to the view controller that is about to be displayed.

The `UIStoryboardSegue` class supports the standard visual transitions available in UIKit. You can also subclass to define custom transitions between the view controllers in your storyboard file.

You do not create segue objects directly. Instead, the storyboard runtime creates them when it must perform a segue between two view controllers. You can still initiate a segue programmatically using the `performSegueWithIdentifier:sender:` method of `UIViewController` if you want. You might do so to initiate a segue from a source that was added programmatically and therefore not available in Interface Builder.

Subclassing Notes

- Initialization:
 - `init(identifier:source:destination:)`
- Accessing attributes:
 - `sourceViewController`
 - `destinationViewController`
 - `identifier`
- Performing:
 - `perform()`
- Creating a Custom Segue

SEGUE

The screenshot shows the Xcode documentation browser for the `UIStoryboardSegue` class. The search bar at the top contains the text "uistorybooksegue". The main content area is titled "UIStoryboardSegue Class Reference". A green callout bubble with the text "Where to?" points to the "Subclassing Notes" section.

UIStoryboardSegue

UIStoryboardPopoverSegue

Google 6

Stack Overflow 6

Initializing a Storyboard Segue

M `init(identifier:source:destination:)`

Accessing the Segue Attributes

P `sourceViewController`

P `destinationViewController`

P `identifier`

Performing the Segue

M `perform()`

Creating a Custom Segue

M `init(identifier:source:destination:)`

UIStoryboardSegue Class Reference

A `UIStoryboardSegue` object is responsible for performing the visual transition between two view controllers. In addition, segue objects are used to prepare for the transition from one view controller to another. Segue objects contain information about the view controllers involved in a transition. When a segue is triggered, but before the visual transition occurs, the storyboard runtime calls the current view controller's `prepareForSegue:sender:` method so that it can pass any needed data to the view controller that is about to be displayed.

The `UIStoryboardSegue` class supports the standard visual transitions available in UIKit. You can also subclass to define custom transitions between the view controllers in your storyboard file.

Instead, the storyboard runtime creates them to view controllers. You can still initiate a `segueWithIdentifier:sender:` method of do so to initiate a segue from a source that are not available in Interface Builder.

Where to?

Subclassing Notes

SEGUE

The screenshot shows the Xcode documentation interface. The title bar says "UIStoryboardSegue Class Reference". The search bar contains the text "uistorybooksegue". The main content area displays the class reference for `UIStoryboardSegue`, which is described as a class responsible for performing visual transitions between view controllers. It supports standard UIKit transitions and can be subclassed for custom transitions. The sidebar on the left lists related topics such as "Initializing a Storyboard Segue", "Accessing the Segue Attributes", "Performing the Segue", and "Creating a Custom Segue".

UIStoryboardSegue Class Reference

A `UIStoryboardSegue` object is responsible for performing the visual transition between two view controllers. In addition, segue objects are used to prepare for the transition from one view controller to another. Segue objects contain information about the view controllers involved in a transition. When a segue is triggered, but before the visual transition occurs, the storyboard runtime calls the current view controller's `prepareForSegue:sender:` method so that it can pass any needed data to the view controller that is about to be displayed.

The `UIStoryboardSegue` class supports the standard visual transitions available in UIKit. You can also subclass to define custom transitions between the view controllers in your storyboard file.

You do not create segue objects directly. Instead, the storyboard runtime creates them when it must perform a segue between two view controllers. You can still initiate a segue programmatically using the `performSegueWithIdentifier:sender:` method of `UIViewController` if you want. You might do so to initiate a segue from a source that was added programmatically and therefore not available in Interface Builder.

Subclassing Notes

- Initialization:
 - `init(identifier:source:destination:)`
- Accessing attributes:
 - `sourceViewController`
 - `destinationViewController`
 - `identifier`
- Performing:
 - `perform()`
- Creating a Custom Segue

SEGUE

The screenshot shows the Xcode documentation interface for the `UIStoryboardSegue` class. The search bar at the top contains the text "uistorybooksegue". The main content area is titled "UIStoryboardSegue Class Reference". The description states:

A `UIStoryboardSegue` object is responsible for performing the visual transition between two view controllers. In addition, segue objects are used to prepare for the transition from one view controller to another. Segue objects contain information about the view controllers involved in a transition. When a segue is triggered, but before the visual transition occurs, the storyboard runtime calls the current view controller's `prepareForSegue:sender:` method so that it can pass any needed data to the view controller that is about to be displayed.

The `UIStoryboardSegue` class supports the standard visual transitions available in UIKit. You can also subclass to define custom transitions between the view controllers in your storyboard file.

On the left sidebar, there is a list of sections and methods:

- Initializing a Storyboard Segue
 - `M init(identifier:source:destination:)`
- Accessing the Segue Attributes
 - `P sourceViewController`
 - `P destinationViewController`
 - `P identifier`
- Performing the Segue
 - `M perform()`
- Creating a Custom Segue
 - `M init(identifier:source:destination:)`

A green callout box with a white border and rounded corners is overlaid on the text, containing the question: "What's its name? Since you can have multiple segues from A to B". Below the callout, the text "Subclassing Notes" is visible.

SEGUE

The screenshot shows the Xcode interface with the following details:

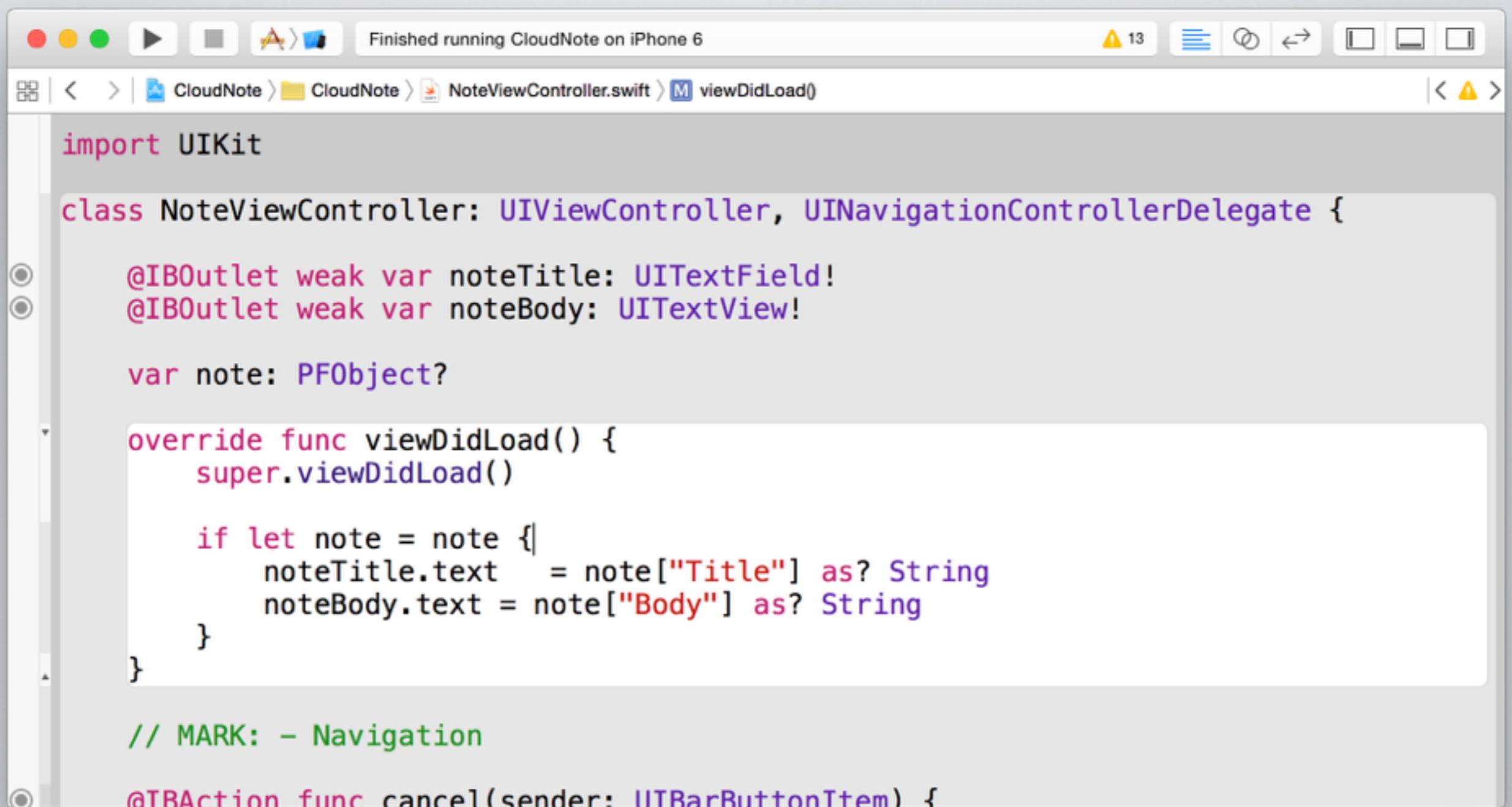
- Toolbar:** Standard Xcode toolbar with icons for running, stopping, and pausing the application.
- Status Bar:** Shows "Finished running CloudNote on iPhone 6".
- Navigation Bar:** Shows the project structure: CloudNote > CloudNote > NoteTableViewController.swift > unwindToNoteList(_:)
- Code Editor:** Displays Swift code for handling segue logic in NoteTableViewController.swift. The code includes methods for preparing segues and unwinding them.

```
// In a storyboard-based application, you will often want to do a little
// preparation before navigation
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if segue.identifier == "ShowDetail" {
        let noteViewController = segue.destinationViewController as!
        NoteViewController

        // Get the cell that generated this segue.
        if let selectedNoteCell = sender as? NoteCell {
            let indexPath = tableView.indexPathForCell(selectedNoteCell)!
            let selectedNote = notes[indexPath.row]
            noteViewController.note = selectedNote
        }
    } else if segue.identifier == "AddItem" {
        print("Adding new note.")
    }
}

@IBAction func unwindToNoteList(sender: UIStoryboardSegue) {
    if let noteViewController = sender.sourceViewController as?
```

SEGUE



A screenshot of the Xcode IDE interface. The title bar shows "Finished running CloudNote on iPhone 6" and has standard OS X window controls. The navigation bar displays the project structure: CloudNote > CloudNote > NoteViewController.swift > viewDidLoad(). It also includes build status (13 warnings), file navigation, and other developer tools. The main editor area contains the following Swift code:

```
import UIKit

class NoteViewController: UIViewController, UINavigationControllerDelegate {

    @IBOutlet weak var noteTitle: UITextField!
    @IBOutlet weak var noteBody: UITextView!

    var note: PFObject?

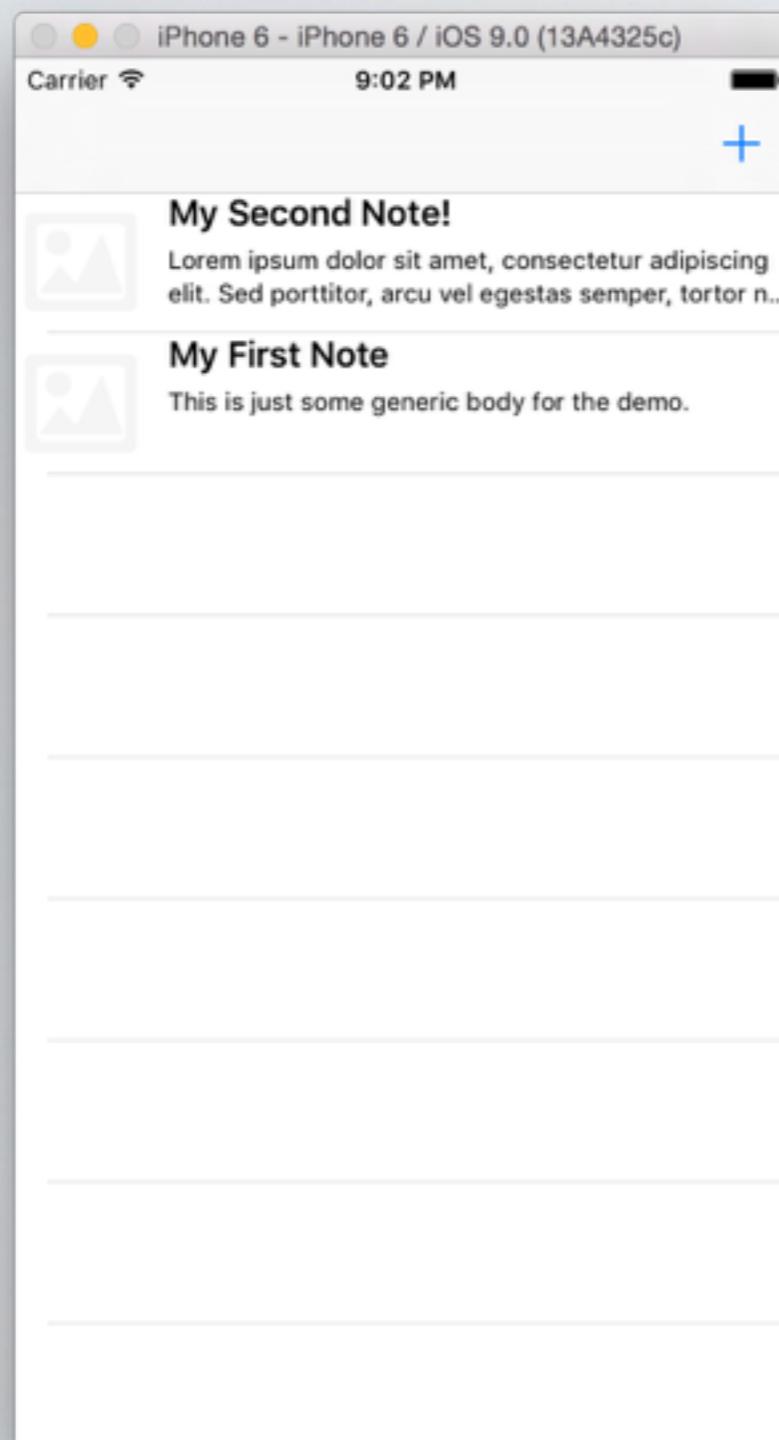
    override func viewDidLoad() {
        super.viewDidLoad()

        if let note = note {
            noteTitle.text = note["Title"] as? String
            noteBody.text = note["Body"] as? String
        }
    }

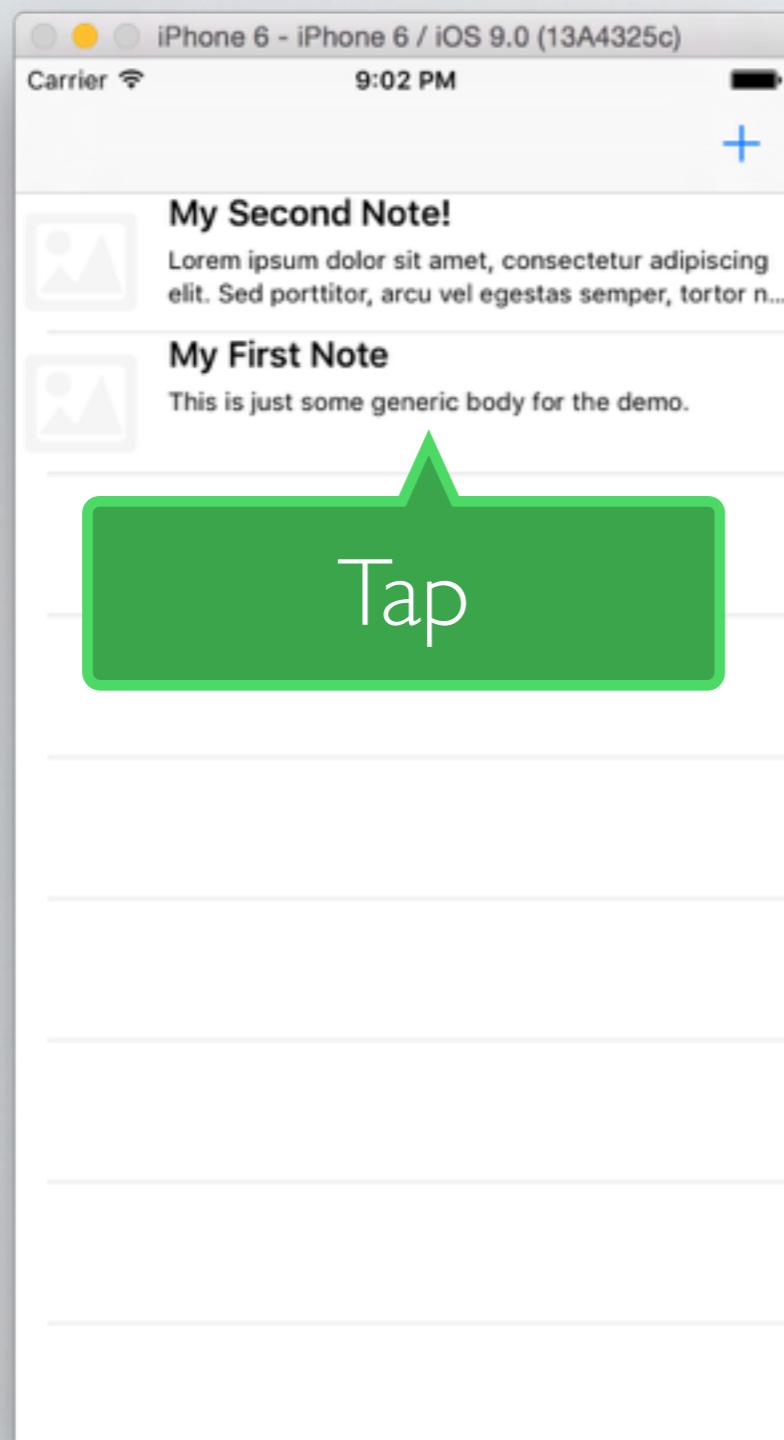
    // MARK: - Navigation

    @IBAction func cancel(sender: UIBarButtonItem) {
```

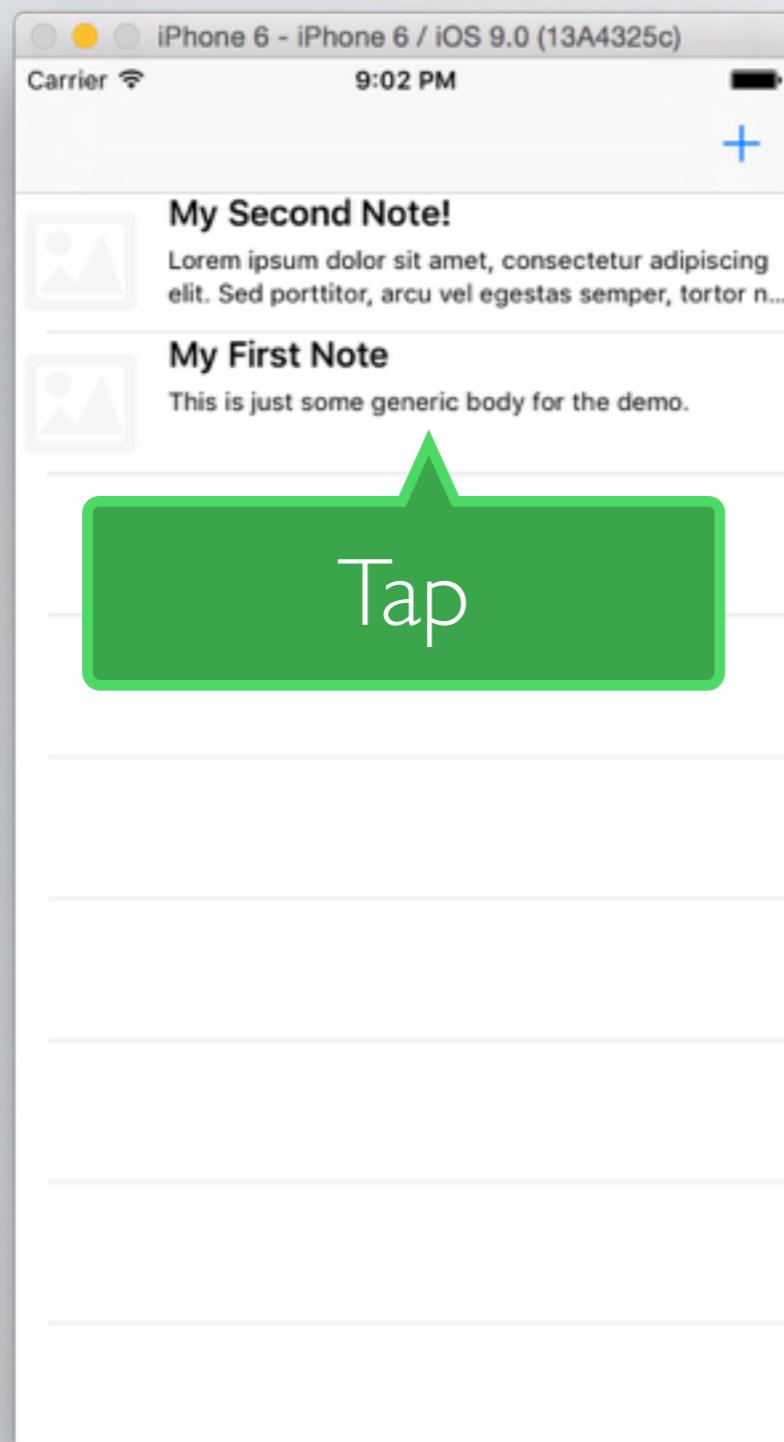
CURRENT VIEW



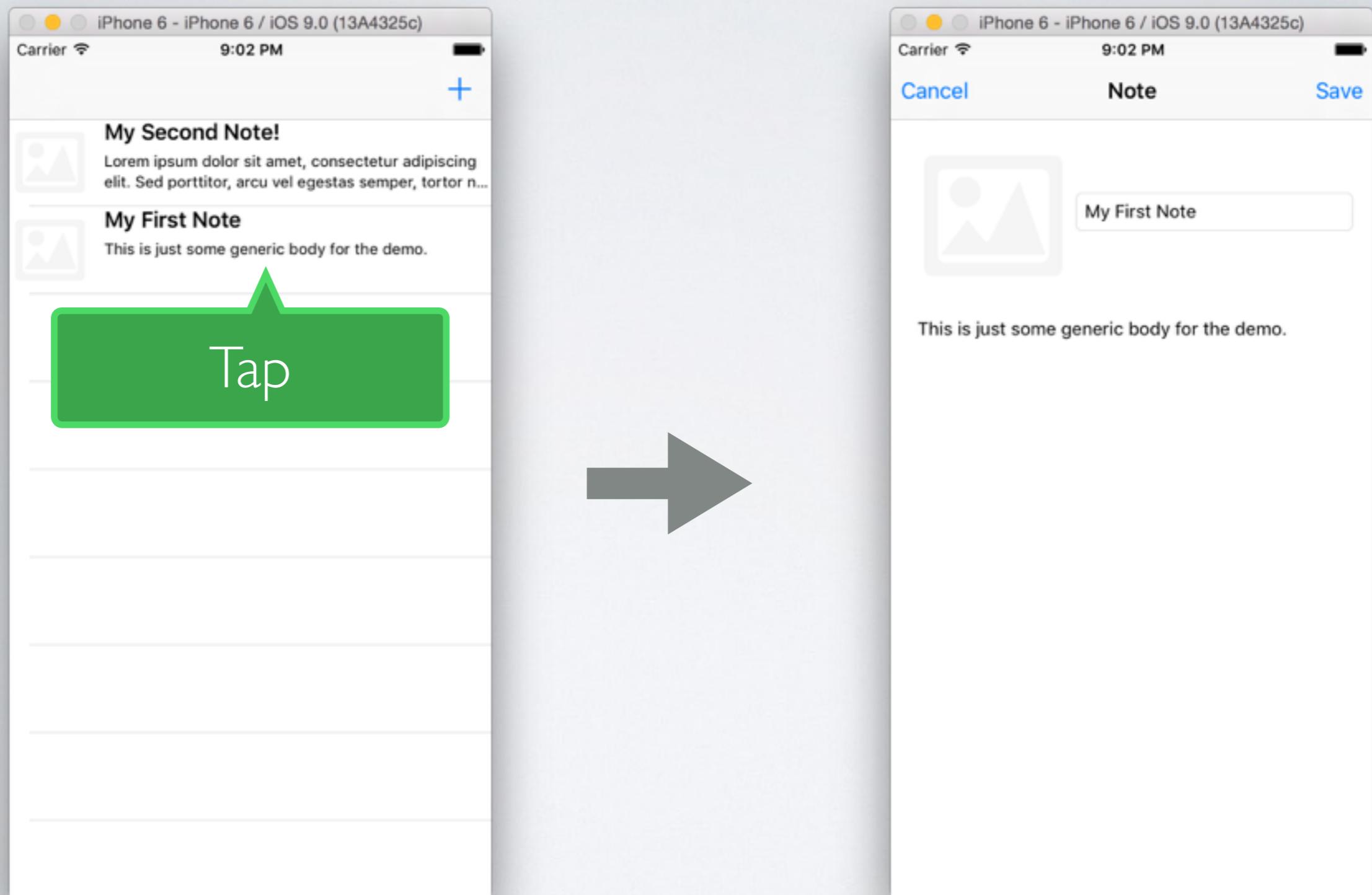
CURRENT VIEW



CURRENT VIEW

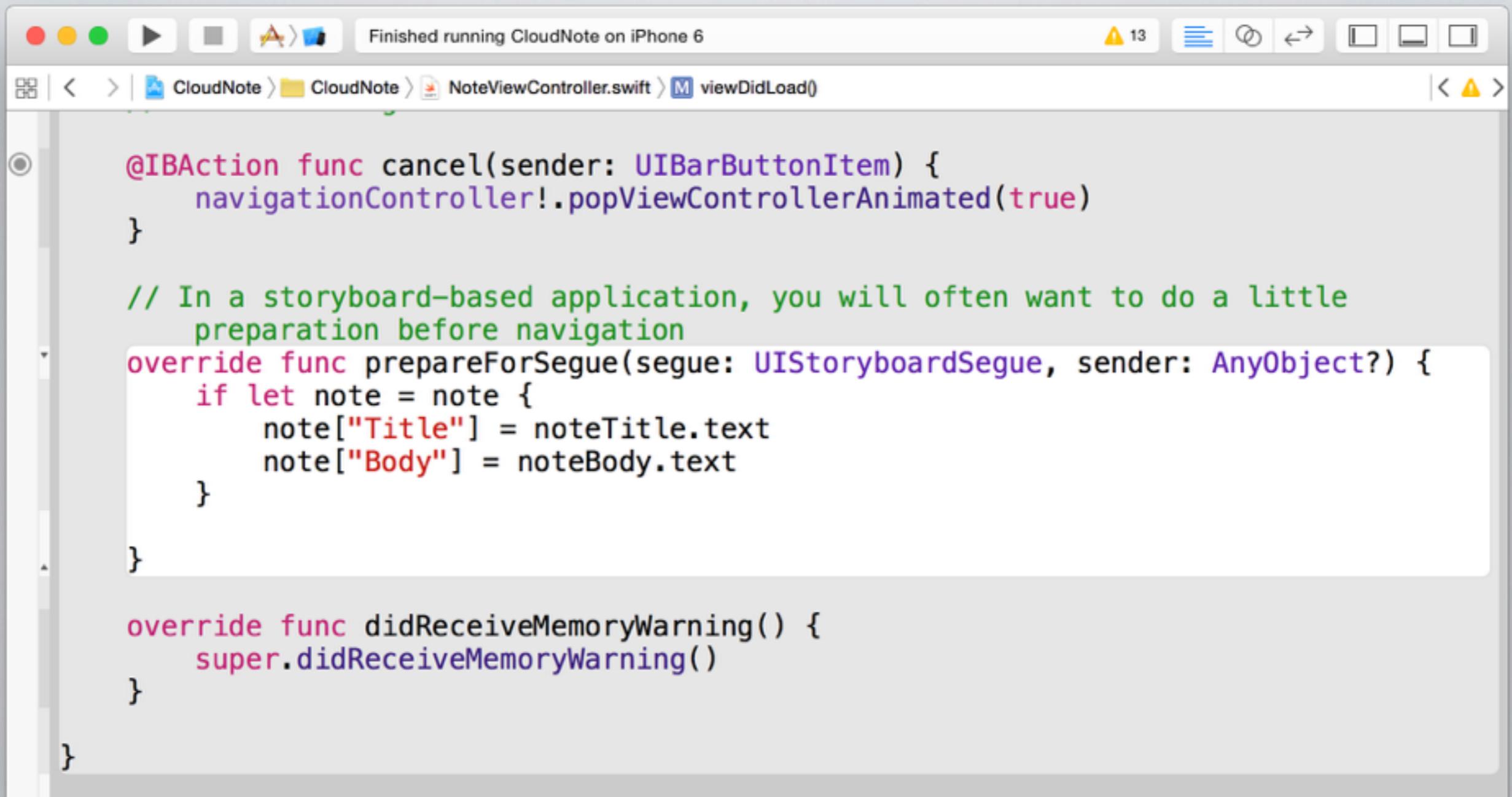


CURRENT VIEW



TASK: UPDATE EXISTING
NOTES

SAVING



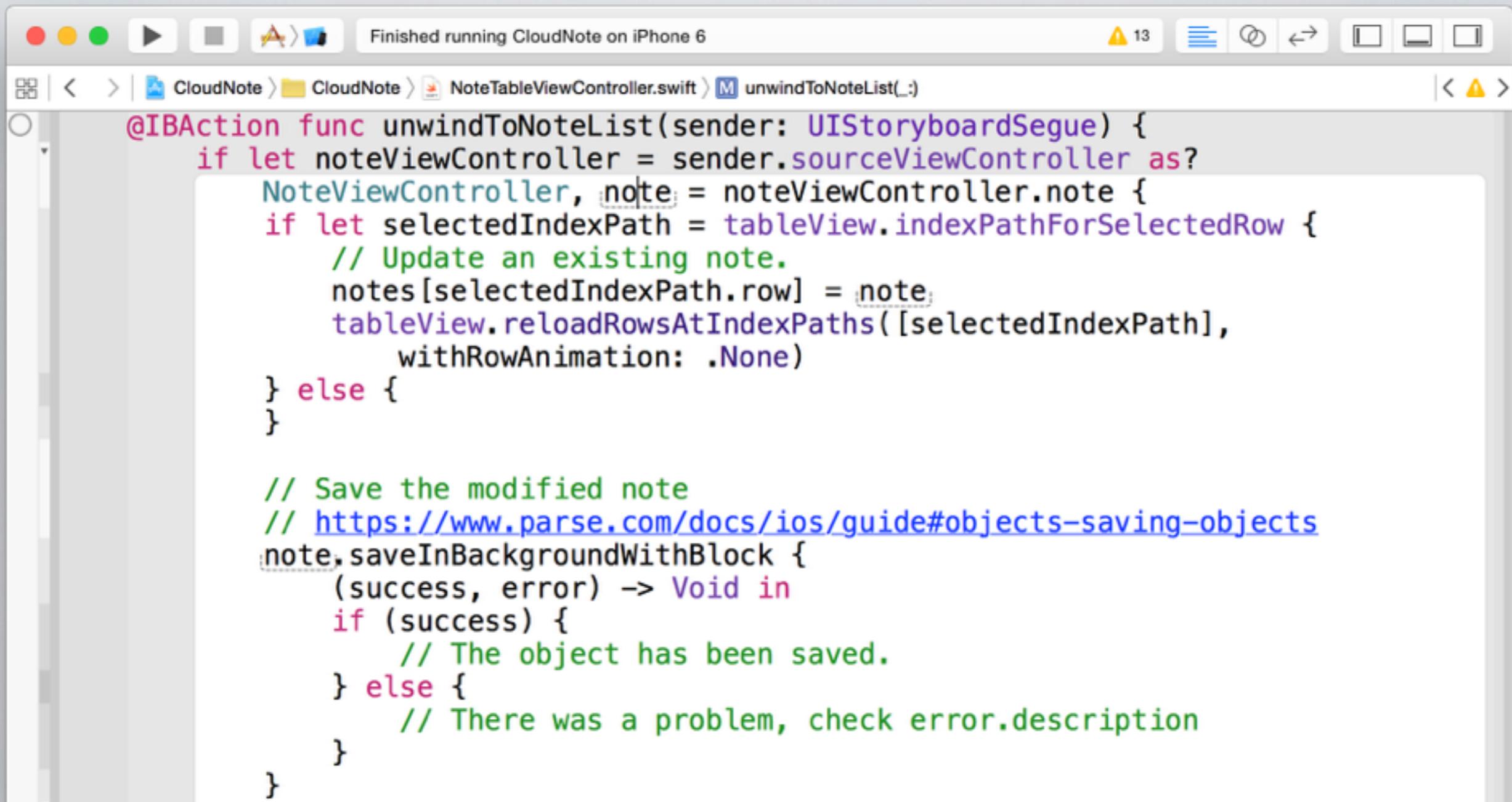
A screenshot of the Xcode IDE interface. The title bar shows "Finished running CloudNote on iPhone 6". The status bar indicates 13 warnings. The navigation bar shows the project structure: CloudNote > CloudNote > NoteViewController.swift > viewDidLoad(). The main editor area displays Swift code for NoteViewController.swift:

```
@IBAction func cancel(sender: UIBarButtonItem) {
    navigationController!.popViewControllerAnimated(true)
}

// In a storyboard-based application, you will often want to do a little
// preparation before navigation
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if let note = note {
        note["Title"] = noteTitle.text
        note["Body"] = noteBody.text
    }
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
}
```

SAVING



The screenshot shows the Xcode IDE interface with the following details:

- Toolbar:** Standard Xcode toolbar with icons for running, stopping, and navigating.
- Status Bar:** Shows "Finished running CloudNote on iPhone 6".
- File Navigator:** Displays the project structure: CloudNote > CloudNote > NoteTableViewController.swift.
- Text Editor:** Contains Swift code for handling a segue from a note view controller back to the note list. The code updates an existing note if a row was selected, or saves a new note if no row was selected. It uses Parse's saveInBackgroundWithBlock method to save the note.

```
@IBAction func unwindToNoteList(sender: UIStoryboardSegue) {
    if let noteViewController = sender.sourceViewController as? NoteViewController, note = noteViewController.note {
        if let indexPath = tableView.indexPathForSelectedRow {
            // Update an existing note.
            notes[indexPath.row] = note
            tableView.reloadRowsAtIndexPaths([indexPath],
                withRowAnimation: .None)
        } else {
        }

        // Save the modified note
        // https://www.parse.com/docs/ios/guide#objects-saving-objects
        note.saveInBackgroundWithBlock {
            (success, error) -> Void in
            if (success) {
                // The object has been saved.
            } else {
                // There was a problem, check error.description
            }
        }
    }
}
```

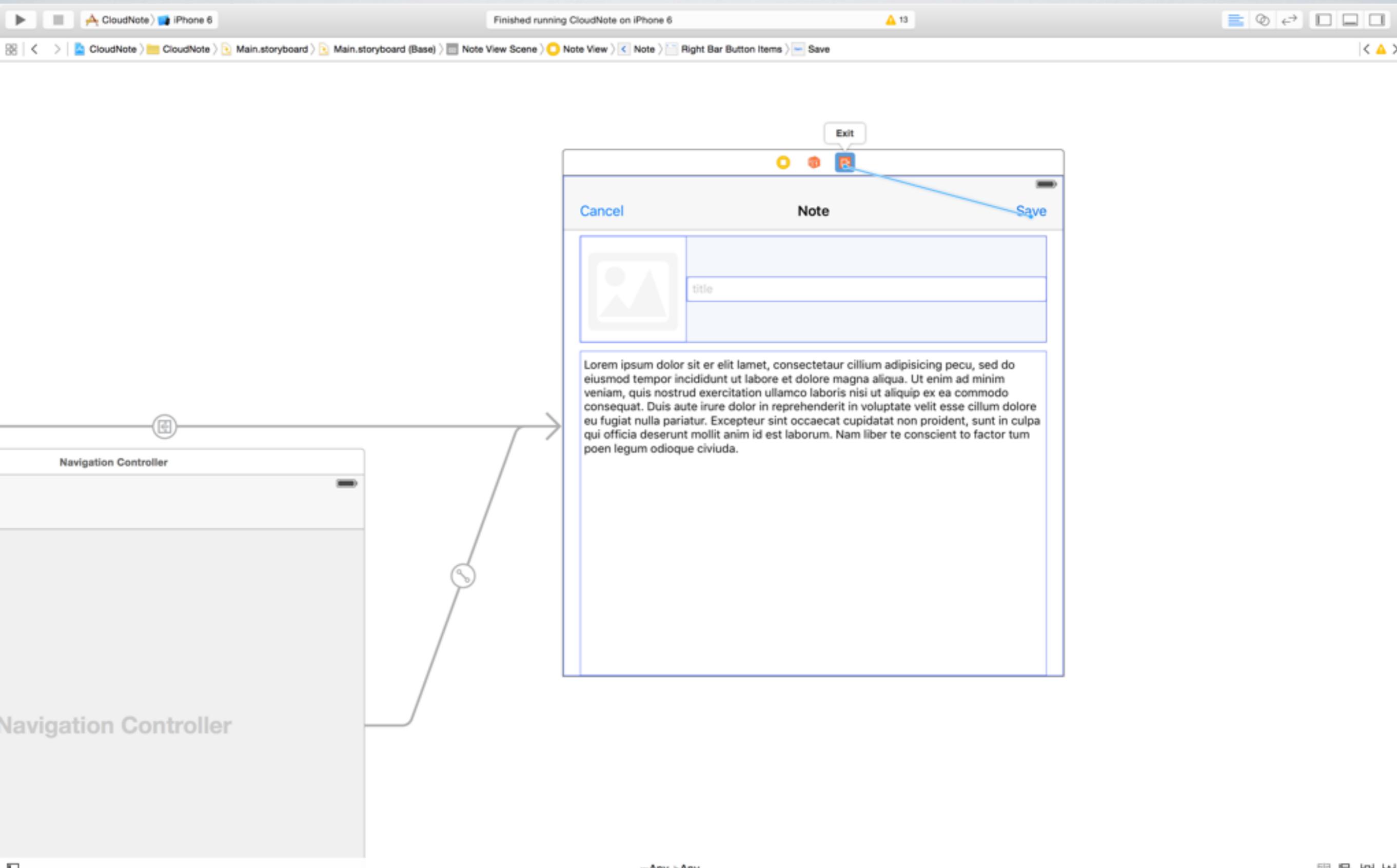
SAVING

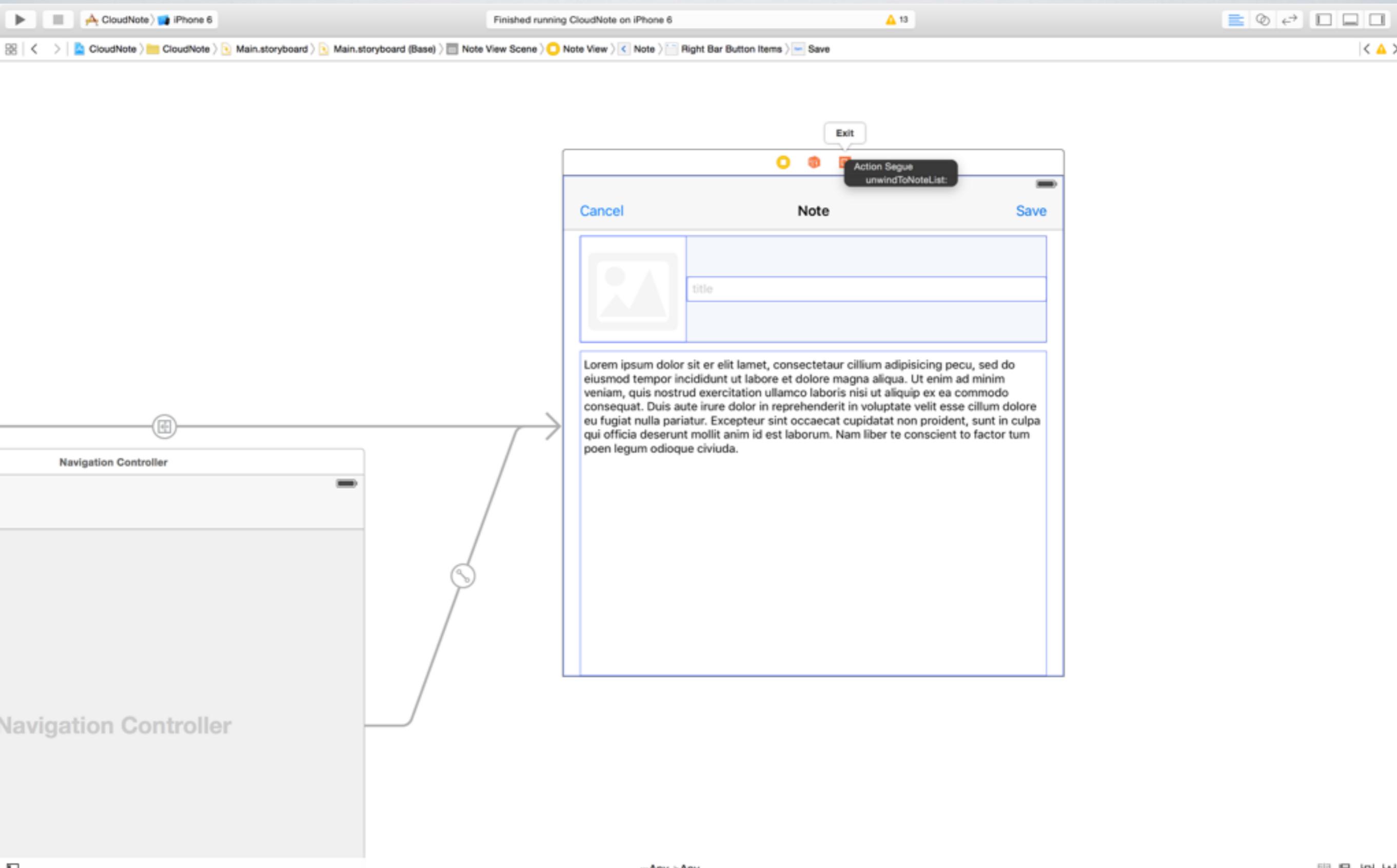


```
Finished running CloudNote on iPhone 6
CloudNote > CloudNote > NoteTableViewController.swift > unwindToNoteList(_:)
@IBAction func unwindToNoteList(sender: UIStoryboardSegue) {
    if let noteViewController = sender.sourceViewController as?
        NoteViewController, note = noteViewController.note {
        if let indexPath = tableView.indexPathForSelectedRow {
            // Update an existing note.
            notes[indexPath.row] = note
            tableView.reloadRowsAtIndexPaths([indexPath],
                withRowAnimation: .None)
        } else {
        }

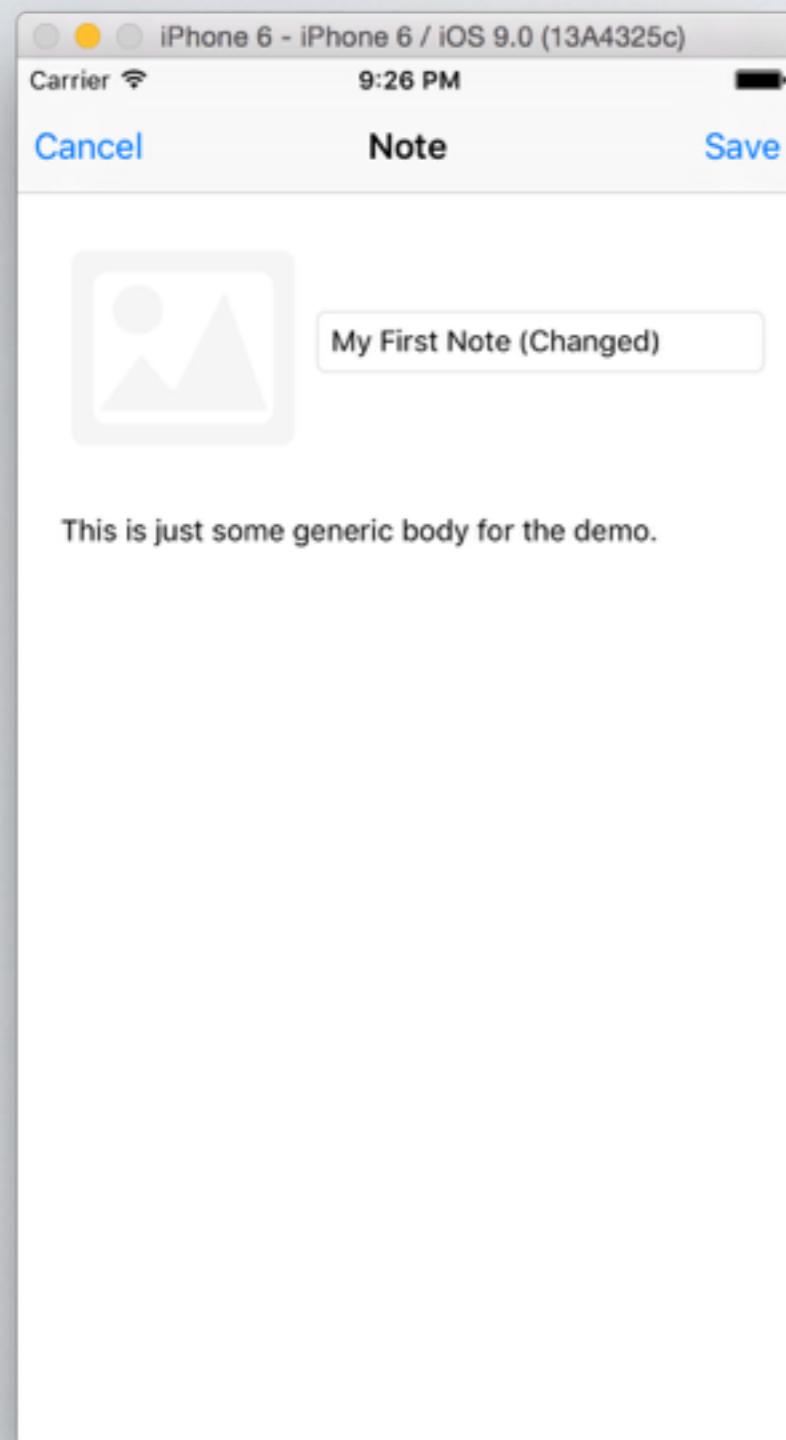
        // Save the modified note
        // https://www.parse.com/docs/ios/guide#objects-saving-objects
        note.saveInBackgroundWithBlock {
            (success, error) -> Void in
            if success {
                print("Note has been saved.")
            } else {
                print("There was an error saving the note: \(error?.description ?? "Unknown error")")
            }
        }
    }
}
```

Saves it back to Parse

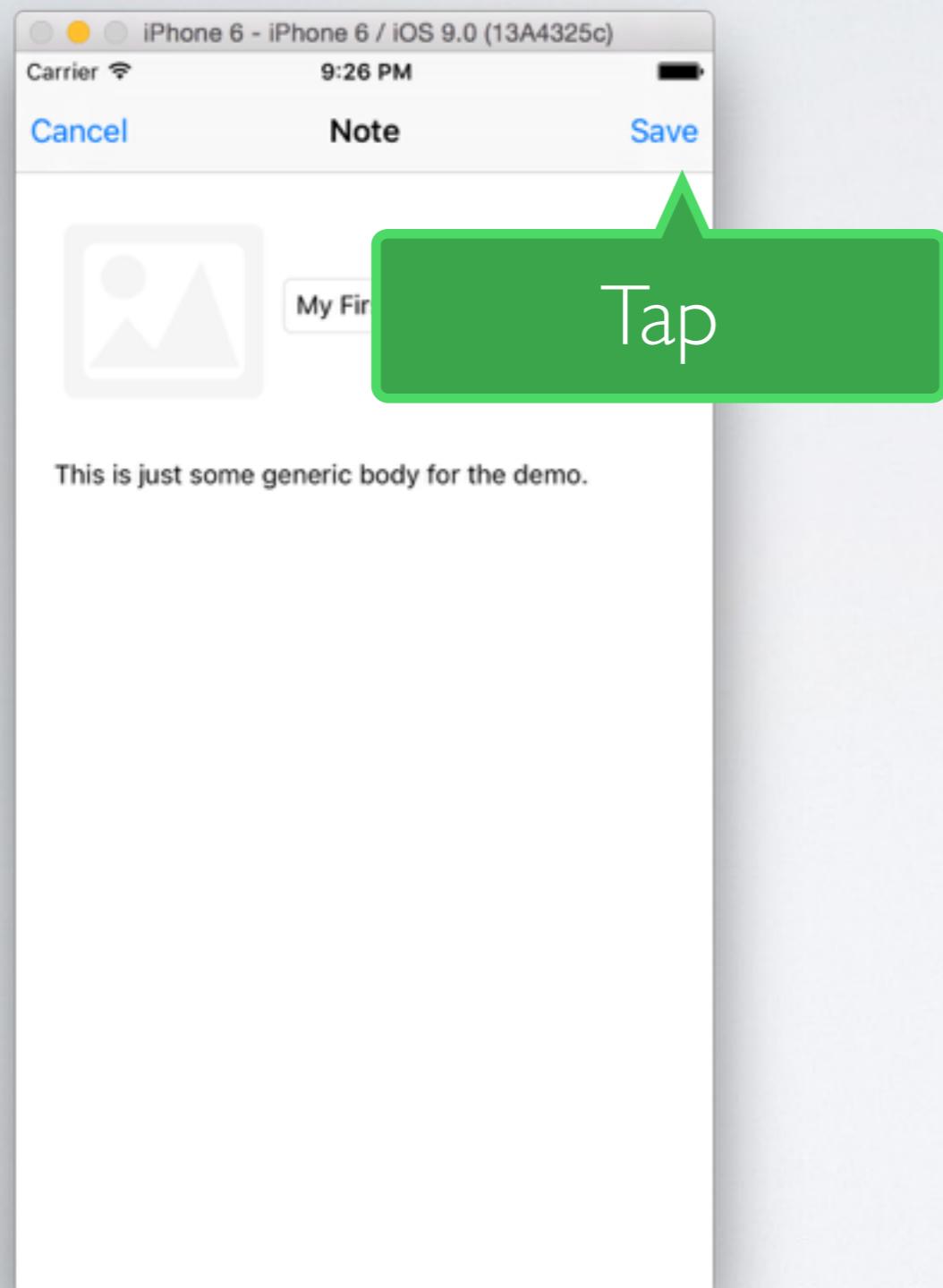




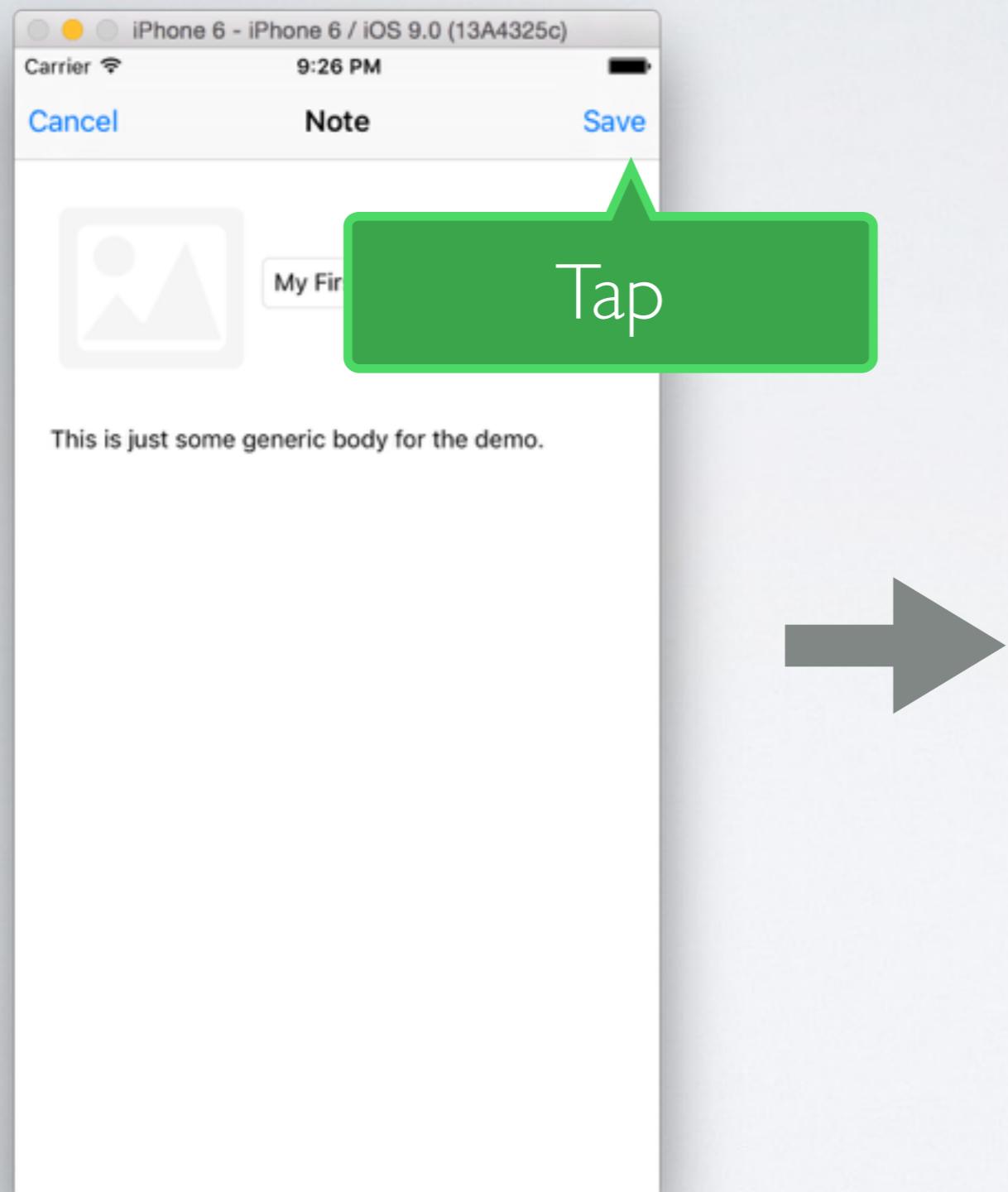
CURRENT VIEW



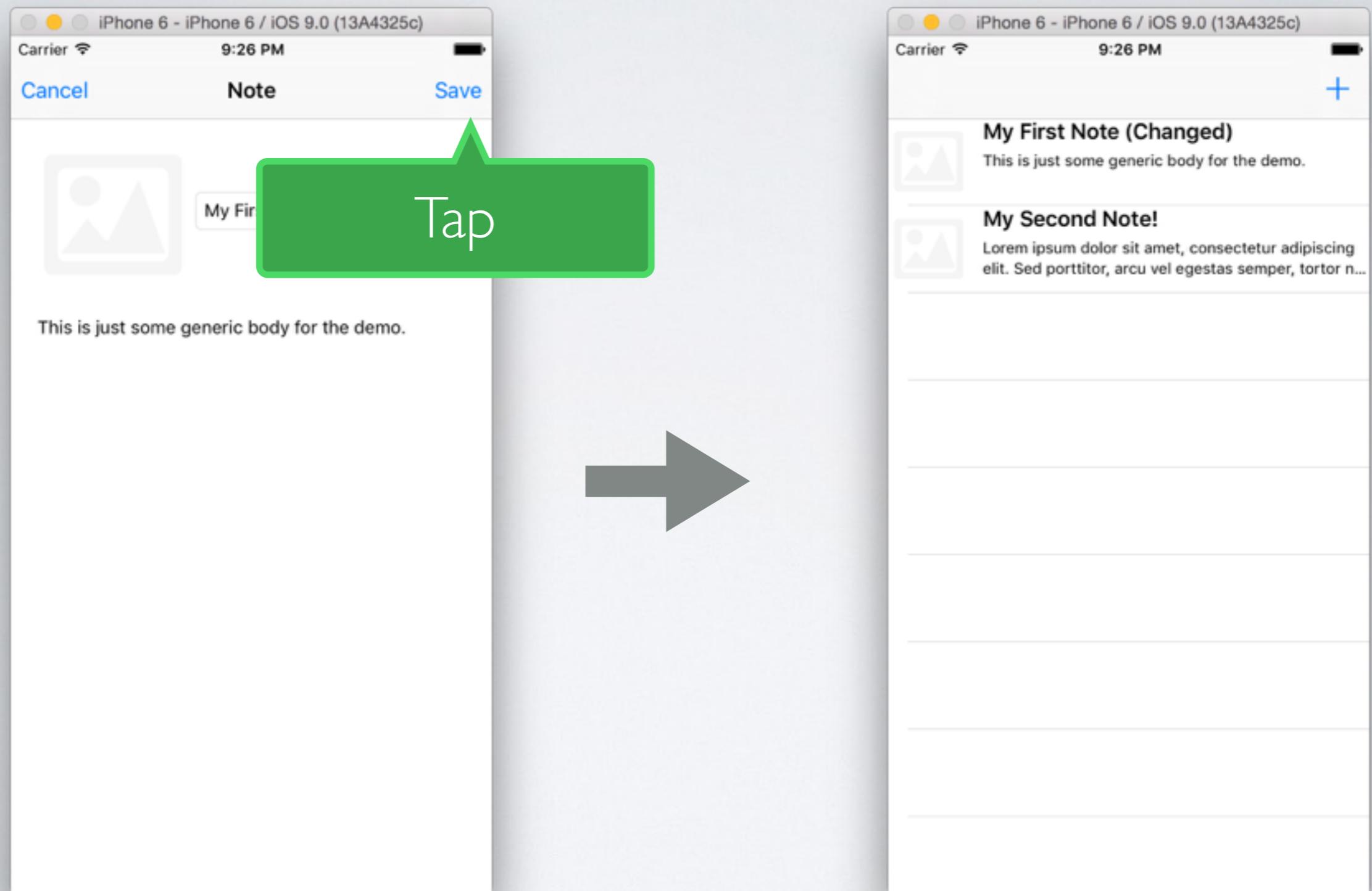
CURRENT VIEW



CURRENT VIEW



CURRENT VIEW



CHANGES ON PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a navigation bar with standard OS X window controls (red, yellow, green buttons), a title bar with the URL "www.parse.com/apps/cloudnote--4/collections", and a toolbar with icons for Core, Security, and other settings.

The main area is titled "CloudNote" and "DEV". On the left, a sidebar menu includes "Data" (selected), "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config".

The "Data" section displays a table for the "Note" class, which has 2 rows. The columns are "objectId", "Title", "createdAt", and "updatedAt". The data is as follows:

	objectId	Title	createdAt	updatedAt
<input type="checkbox"/>	SSX7Xq30AK	My Second Note!	Aug 09, 2015, 05:51	Aug 09, 2015, 05:51
<input type="checkbox"/>	ug9aWR71Ud	My First Note (Changed)	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

Below the table are buttons for "+ Add Class", "+ Row", "- Row", "+ Col", "Security", "More", and a filter icon. At the bottom right, there are buttons for "20 rows/page" and a back arrow.

At the very bottom, there are links for "Docs", "Billing", "Downloads", "Help", and "Status".

CHANGES ON PARSE

The screenshot shows the Parse.com interface for the 'CloudNote' application in 'DEV' mode. The left sidebar has 'Data' selected. The main area displays a table for the 'Note' class with two rows. A green callout box with rounded corners contains the text 'Yup, changes are saved'. The table columns are: objectId, Title, createdAt, updatedAt.

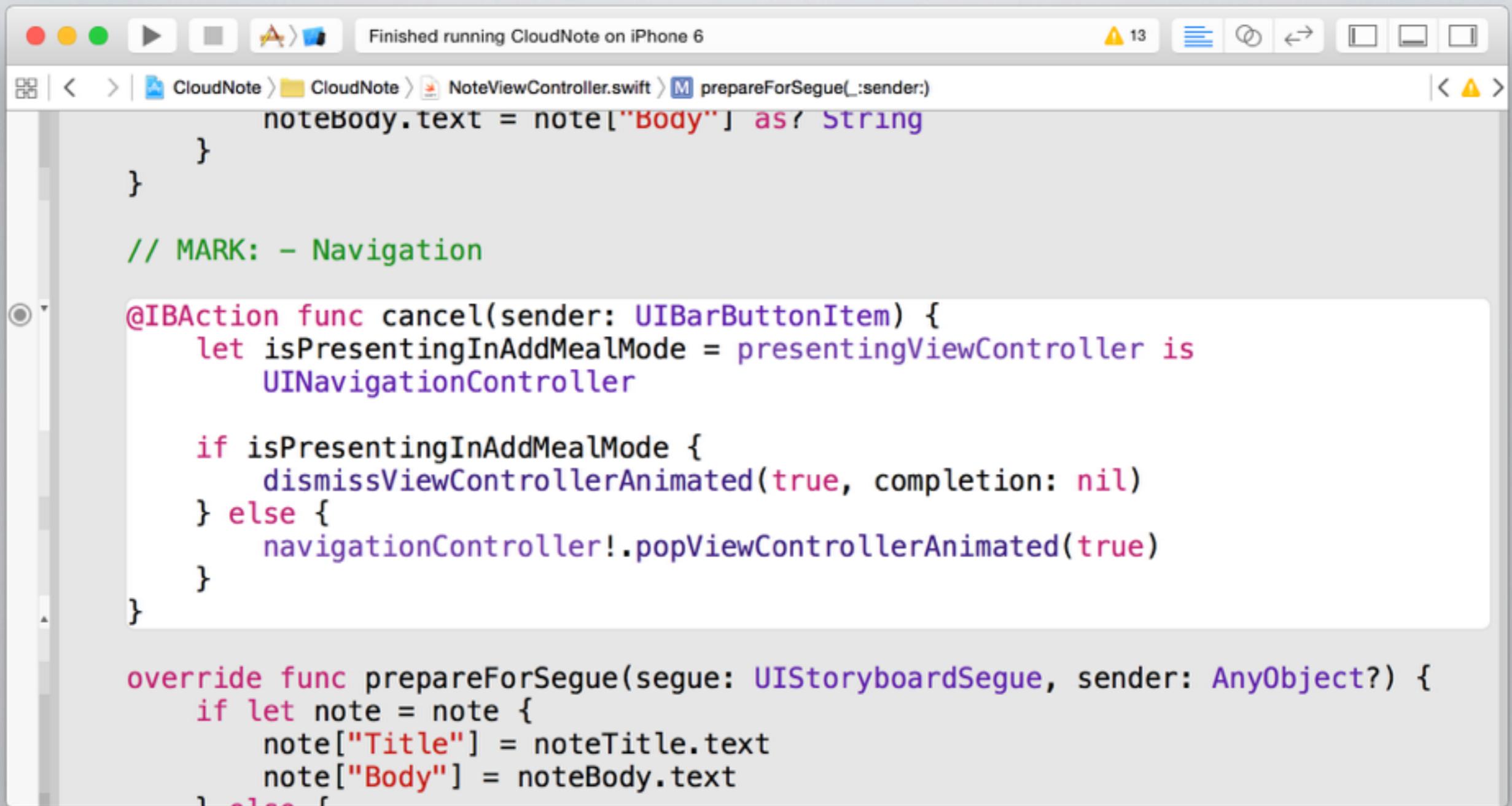
	objectId	Title	createdAt	updatedAt
<input type="checkbox"/>	SSX7Xq30AK	My Second Note!	Aug 09, 2015, 05:51	Aug 09, 2015, 05:51
<input type="checkbox"/>	ug9aWR71Ud	My First Note (Changed)	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

Yup, changes are
saved

Docs Billing Downloads Help Stats

TASK: ADD NEW NOTES

ADDING



The screenshot shows the Xcode interface with the following details:

- Toolbar:** Standard Xcode toolbar with icons for running, stopping, and pausing the application.
- Status Bar:** Shows "Finished running CloudNote on iPhone 6".
- Document Outline:** Shows the project structure: CloudNote > CloudNote > NoteViewController.swift.
- Text Editor:** Displays the code for NoteViewController.swift. The code includes methods for handling cancel and prepareForSegue events, as well as logic for updating a note's title and body.

```
noteBody.text = note["Body"] as? String
}

}

// MARK: - Navigation

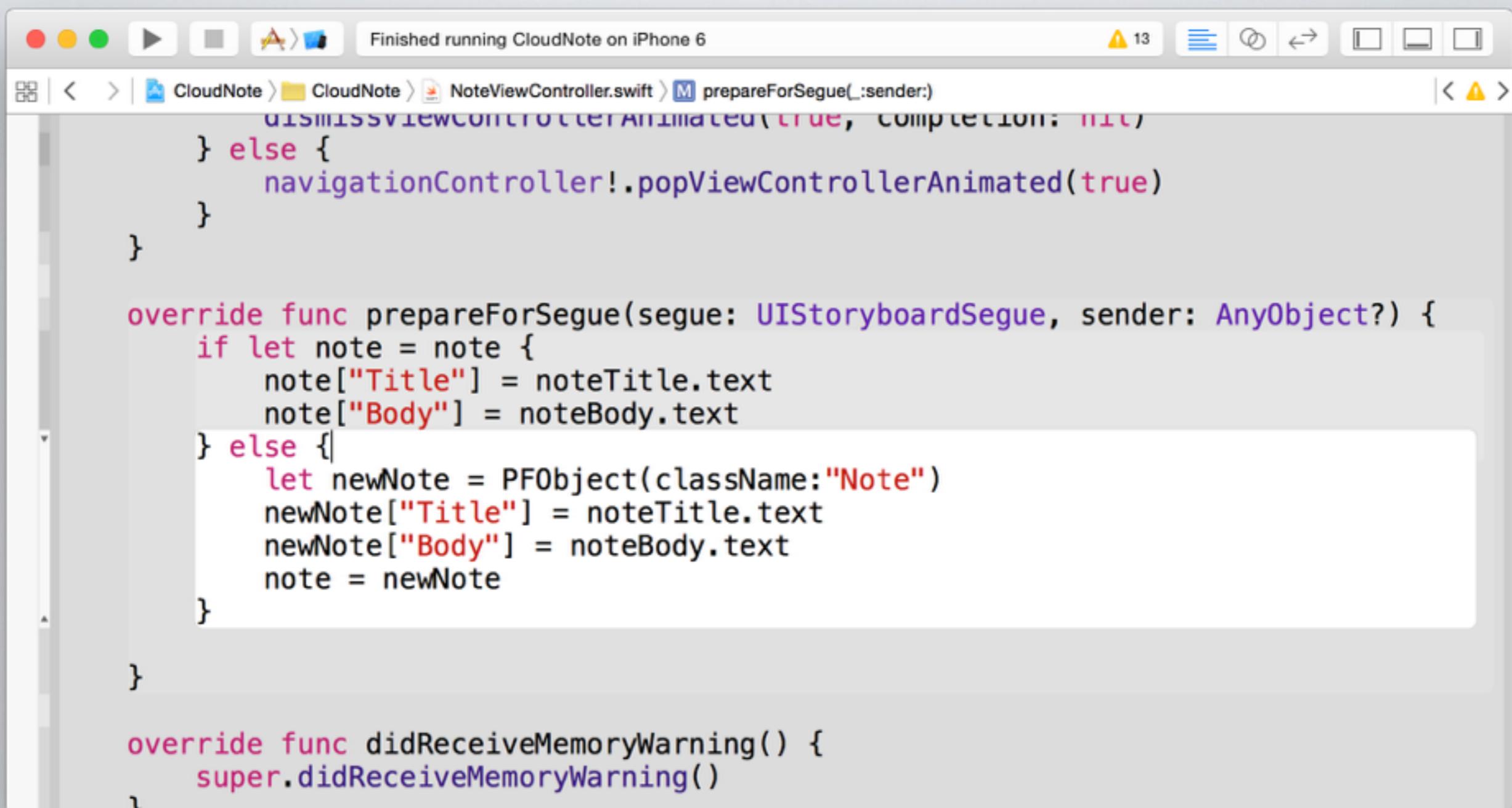
@IBAction func cancel(sender: UIBarButtonItem) {
    let isPresentingInAddMealMode = presentingViewController is
        UINavigationController

    if isPresentingInAddMealMode {
        dismissViewControllerAnimated(true, completion: nil)
    } else {
        navigationController!.popViewControllerAnimated(true)
    }
}

override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if let note = note {
        note["Title"] = noteTitle.text
        note["Body"] = noteBody.text
    } else {

```

ADDING



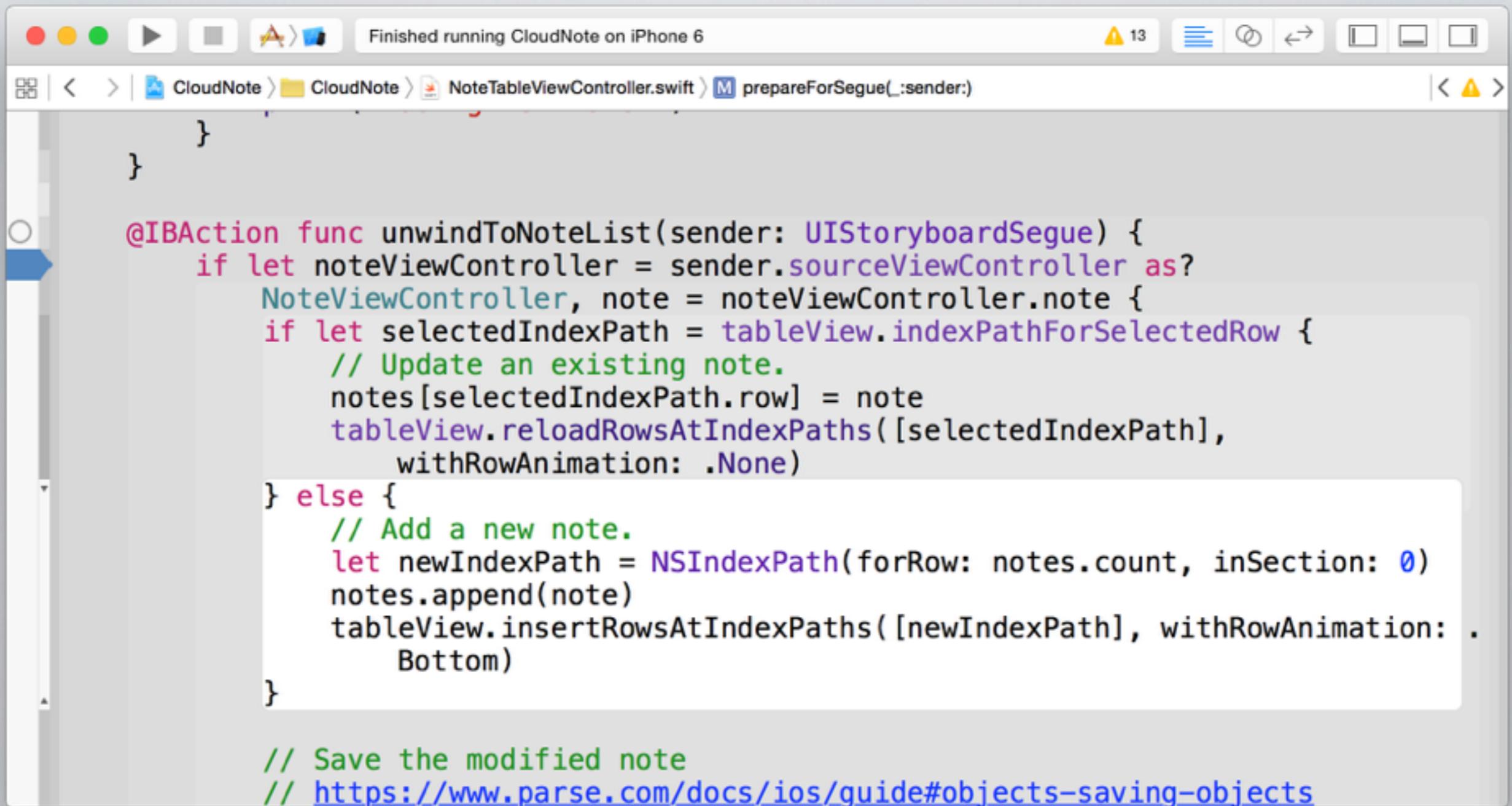
A screenshot of the Xcode IDE showing the code for `NoteViewController.swift`. The window title bar says "Finished running CloudNote on iPhone 6". The status bar shows 13 warnings. The code implements `prepareForSegue(_:sender:)` and `override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?)` to handle note transitions. It also overrides `didReceiveMemoryWarning()`.

```
    navigationController!.popViewControllerAnimated(true)
}
}

override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if let note = note {
        note["Title"] = noteTitle.text
        note["Body"] = noteBody.text
    } else {
        let newNote = PFObject(className:"Note")
        newNote["Title"] = noteTitle.text
        newNote["Body"] = noteBody.text
        note = newNote
    }
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
}
```

ADDING



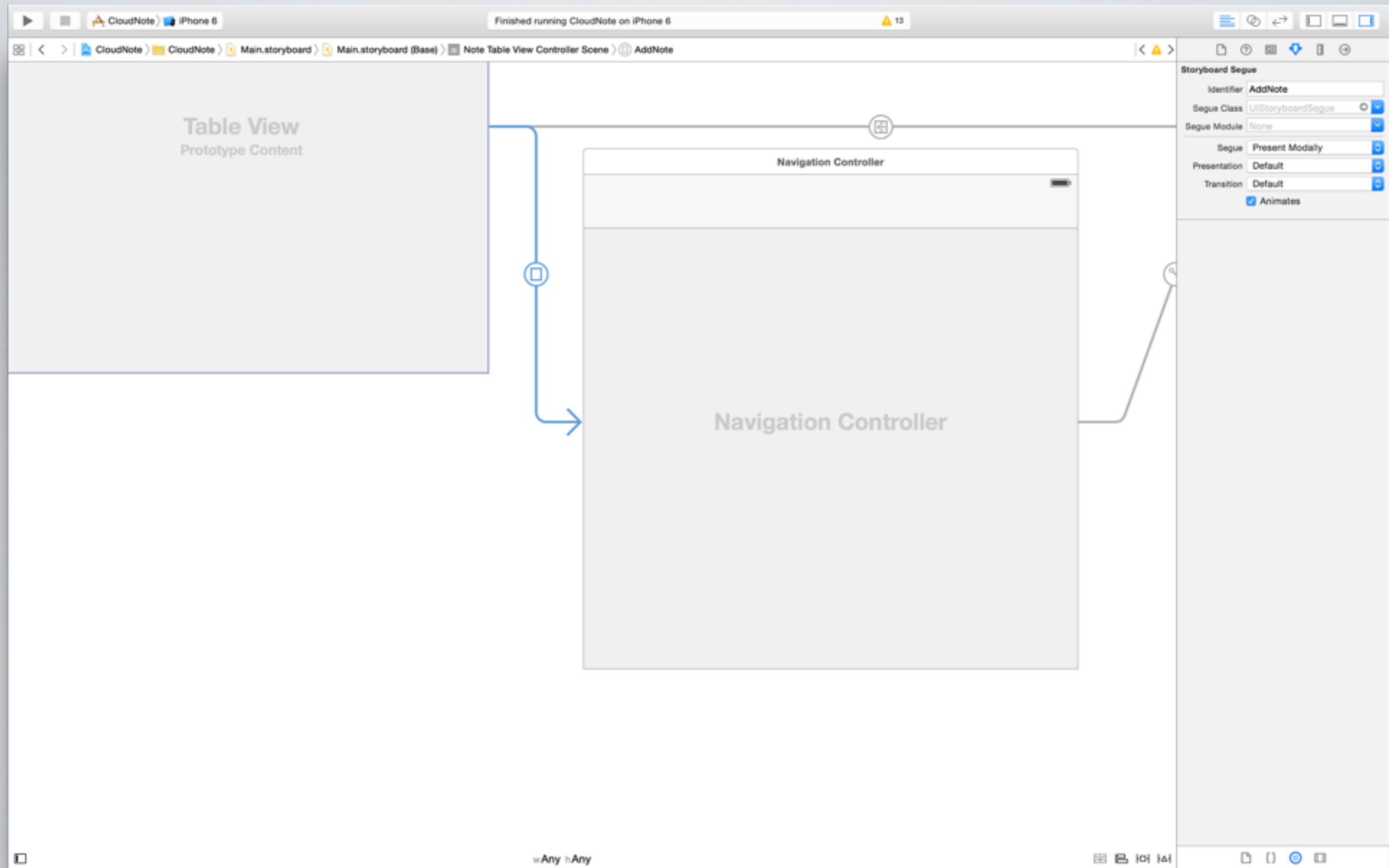
The screenshot shows the Xcode interface with the following details:

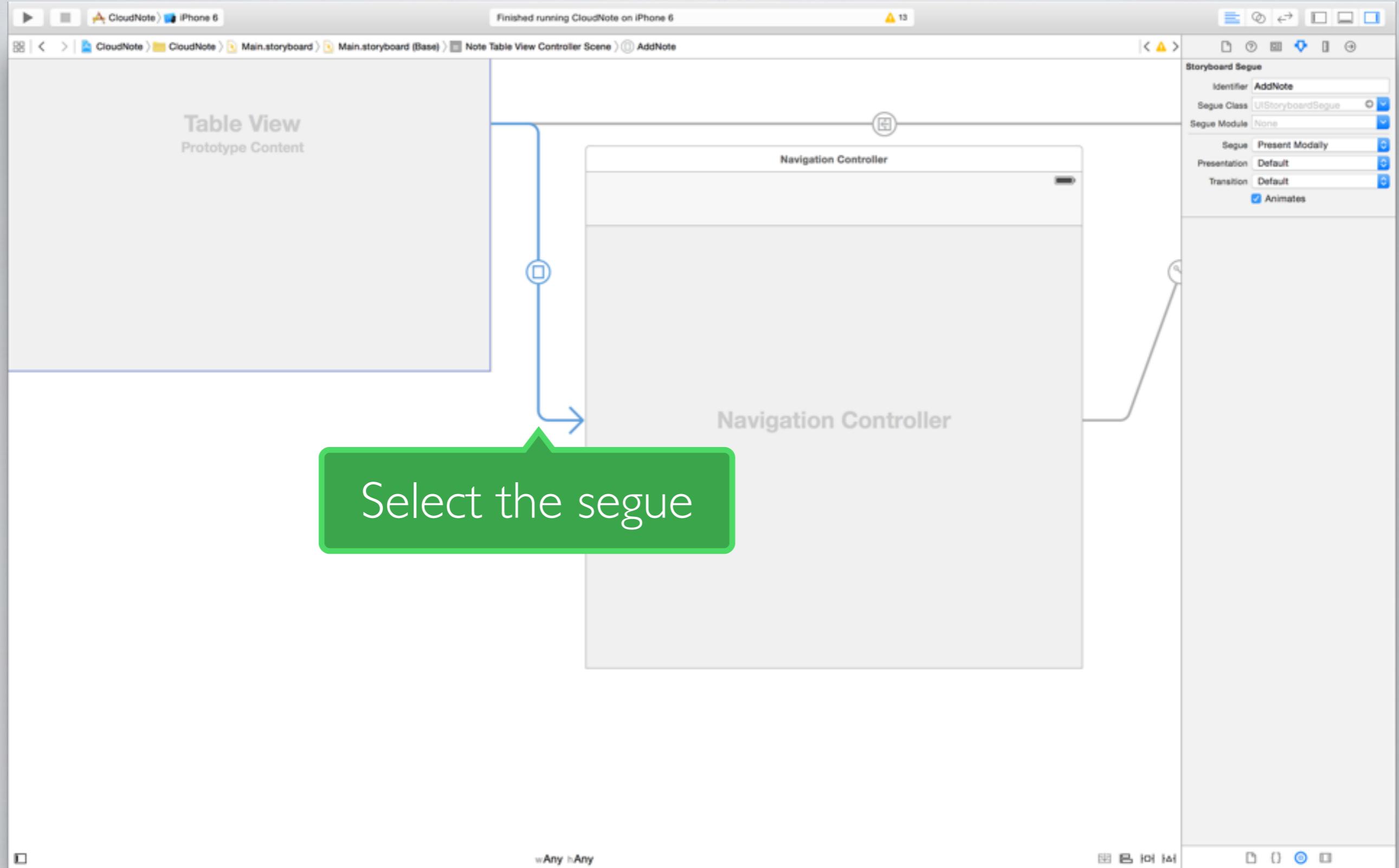
- Toolbar:** Standard Xcode toolbar with icons for running, stopping, and saving.
- Status Bar:** Shows "Finished running CloudNote on iPhone 6" and a warning count of "⚠ 13".
- Project Navigator:** Shows the project structure: CloudNote > CloudNote > NoteTableViewController.swift.
- Editor:** Displays the Swift code for `NoteTableViewController.swift`. The code handles unwind segues from a note view controller back to the note list table view, updating existing notes or adding new ones. It also includes a note about saving modified notes using Parse's iOS guide.

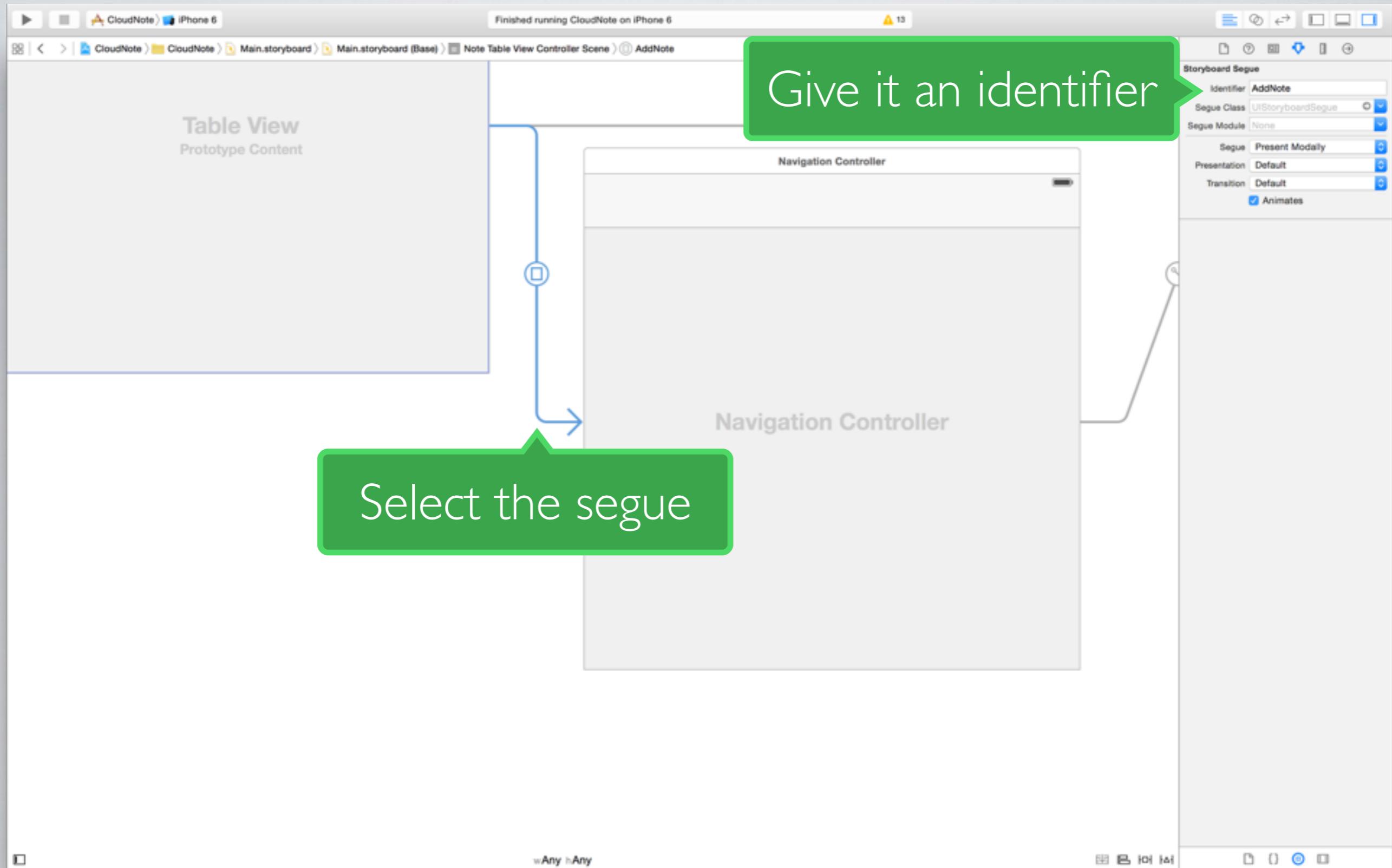
```
        }
    }

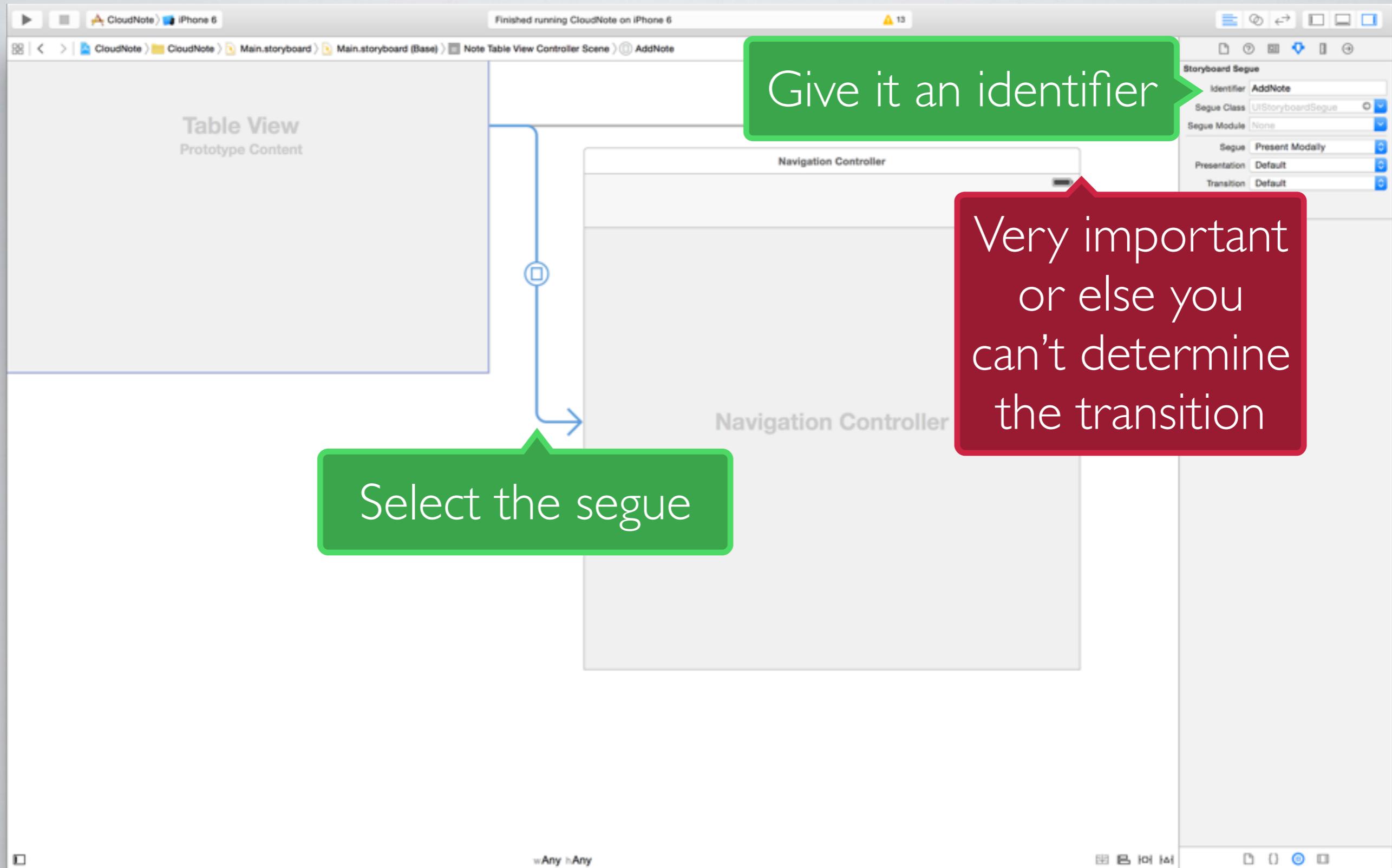
    @IBAction func unwindToNoteList(sender: UIStoryboardSegue) {
        if let noteViewController = sender.sourceViewController as?
            NoteViewController, note = noteViewController.note {
            if let selectedIndexPath = tableView.indexPathForSelectedRow {
                // Update an existing note.
                notes[selectedIndexPath.row] = note
                tableView.reloadRowsAtIndexPaths([selectedIndexPath],
                                                withRowAnimation: .None)
            } else {
                // Add a new note.
                let newIndexPath = NSIndexPath(forRow: notes.count, inSection: 0)
                notes.append(note)
                tableView.insertRowsAtIndexPaths([newIndexPath], withRowAnimation: .
                                                Bottom)
            }
        }
    }

    // Save the modified note
    // https://www.parse.com/docs/ios/guide#objects-saving-objects
}
```

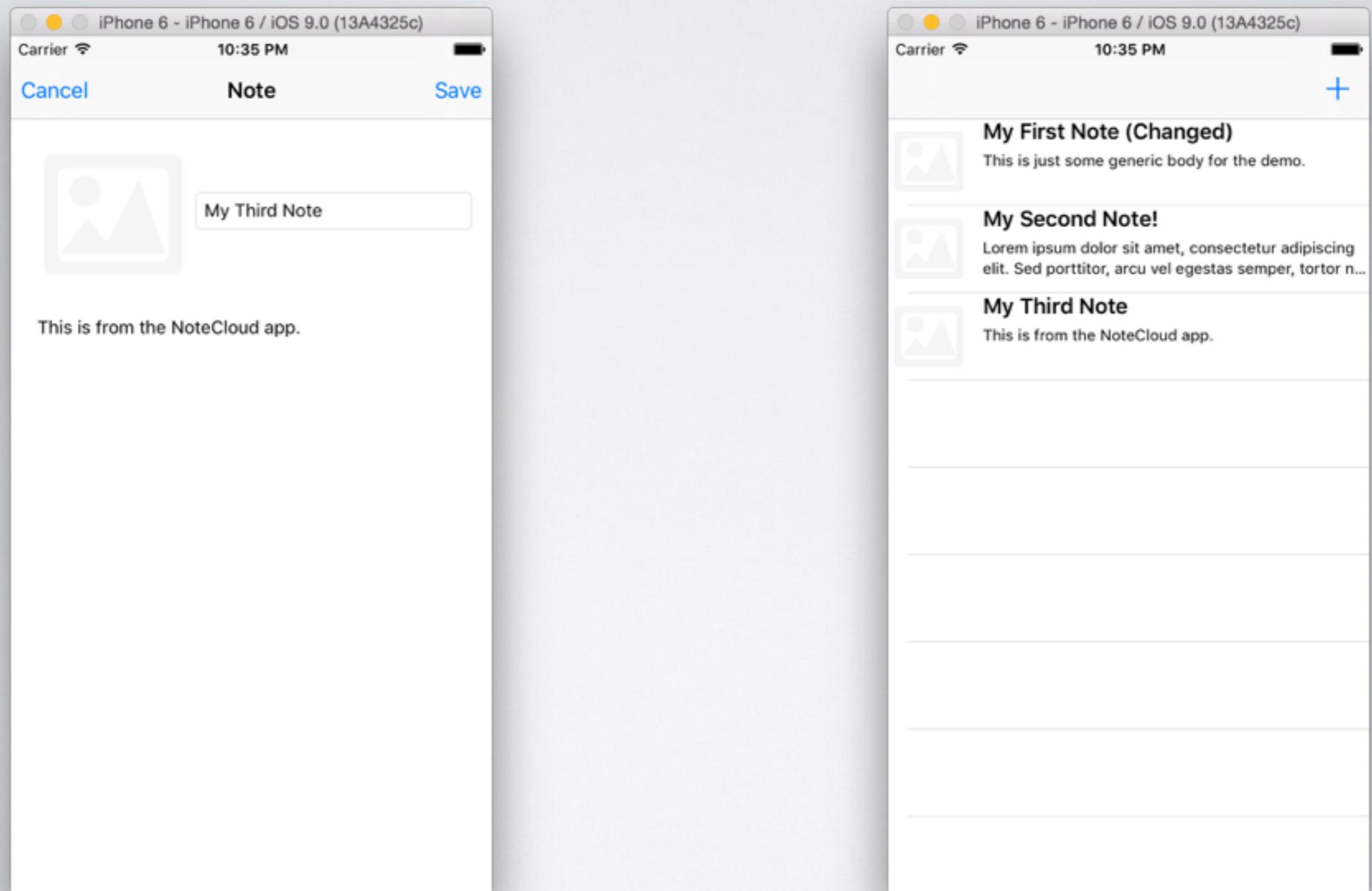




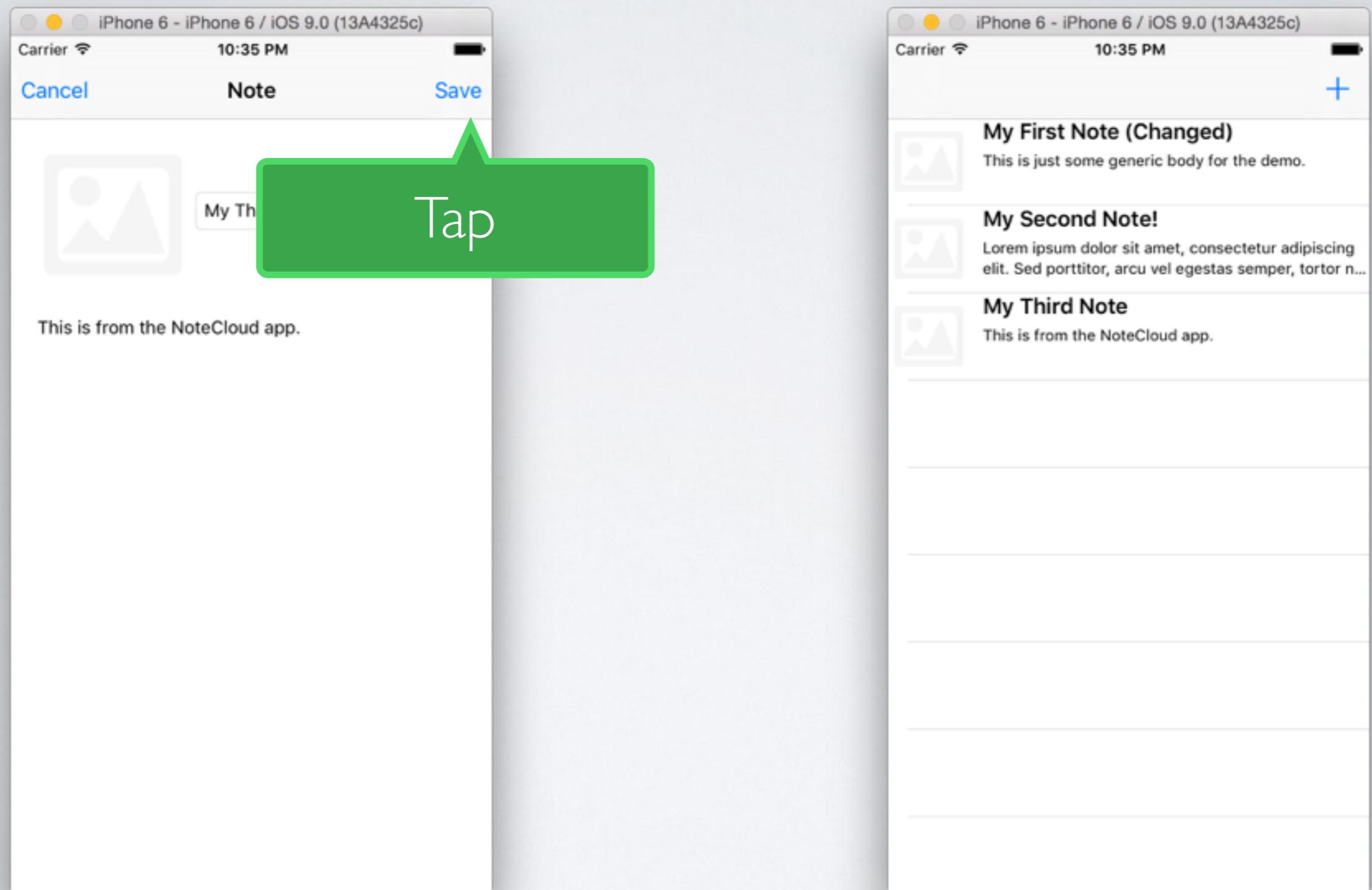




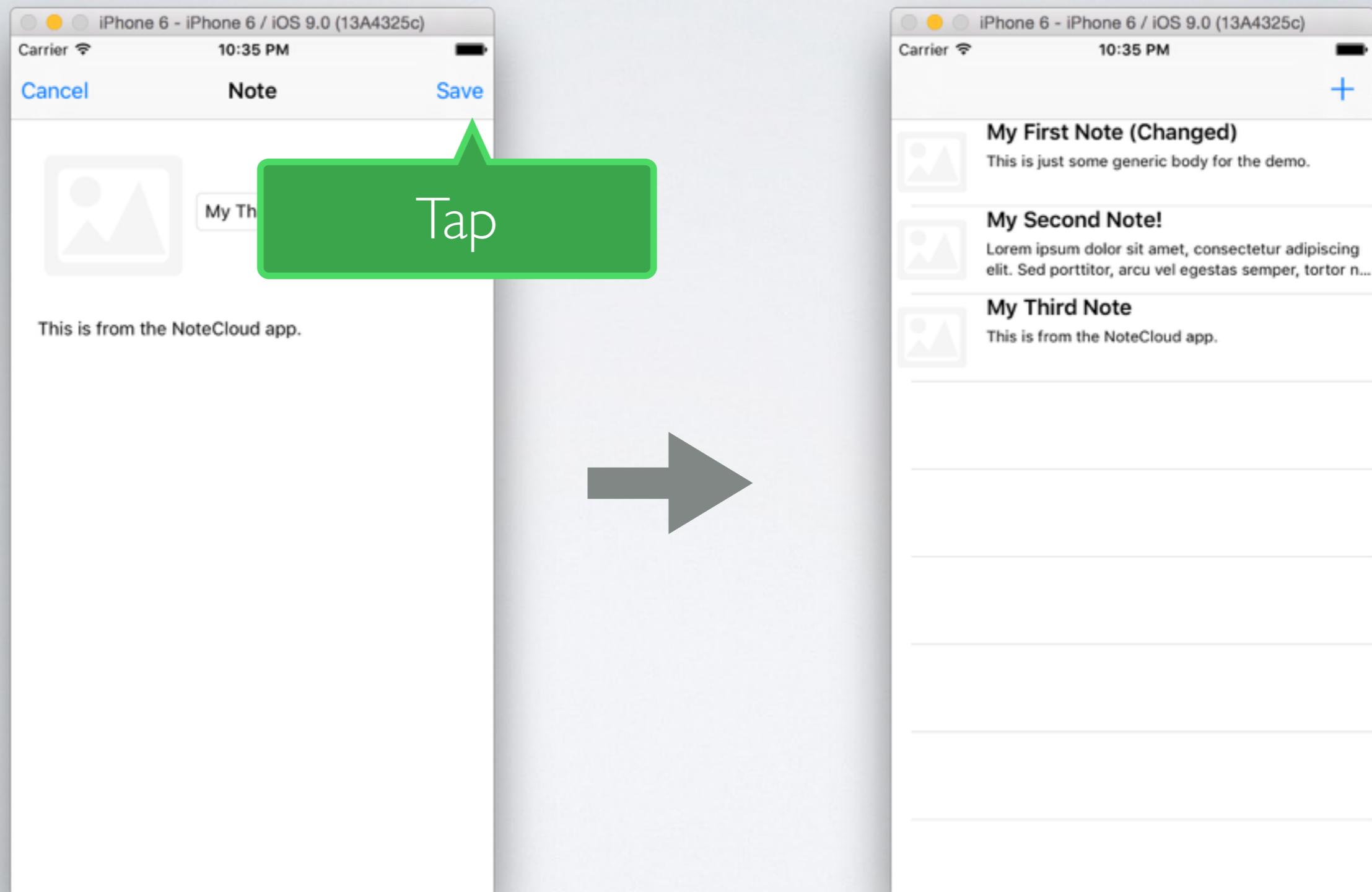
CURRENTVIEW



CURRENTVIEW



CURRENT VIEW



CHANGES ON PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a toolbar with standard OS X window controls (red, yellow, green buttons), a refresh icon, and a URL bar displaying www.parse.com/apps/cloudnote--4/collections. Below the toolbar is a header with the application name "CloudNote" and a "DEV" environment switch. The main navigation bar includes links for "Core", "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config". On the far right of the header is a user profile picture.

The main content area is titled "Data" and displays a table of "Note" objects. The table has columns for "objectId", "Title", "createdAt", and "updatedAt". There are buttons for "+ Row", "- Row", "+ Col", "Security", and "More". A filter icon is also present. The table shows three rows of data:

Note	objectId	Title	createdAt	updatedAt
3	OzR79Cfyuo	My Third Note	Aug 11, 2015, 05:35	Aug 11, 2015, 05:35
	SSX7Xq30AK	My Second Note!	Aug 09, 2015, 05:51	Aug 09, 2015, 05:51
	ug9aWR71Ud	My First Note (Changed)	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

On the left sidebar, there are buttons for "+ Add Class" and "Import". The bottom right corner of the dashboard shows pagination controls for "20 rows/page" and a back/forward navigation icon.

CHANGES ON PARSE

The screenshot shows the Parse.com dashboard for the 'CloudNote' application in 'DEV' mode. The 'Data' tab is selected, displaying a table of notes. A green callout box with the text 'Yup, new note is created' points to the fourth row of the table, which represents a newly added note.

Note	objectId	Title	createdAt	updatedAt
3	OzR79Cfyuo	My Third Note	Aug 11, 2015, 05:35	Aug 11, 2015, 05:35
	SSX7Xq30AK	My Second Note!	Aug 09, 2015, 05:51	Aug 09, 2015, 05:51
	ug9aWR73		Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

Yup, new note is created

TASK: DELETE EXISTING
NOTES

DELETING OBJECT API

The screenshot shows a web browser window with the URL www.parse.com/docs/ios/guide#obje. The page is titled "Getting Started" and features a section titled "Deleting Objects". It contains sample code for Objective-C and Swift, and descriptive text about deleting objects from the cloud.

Getting Started

Deleting Objects

To delete an object from the cloud:

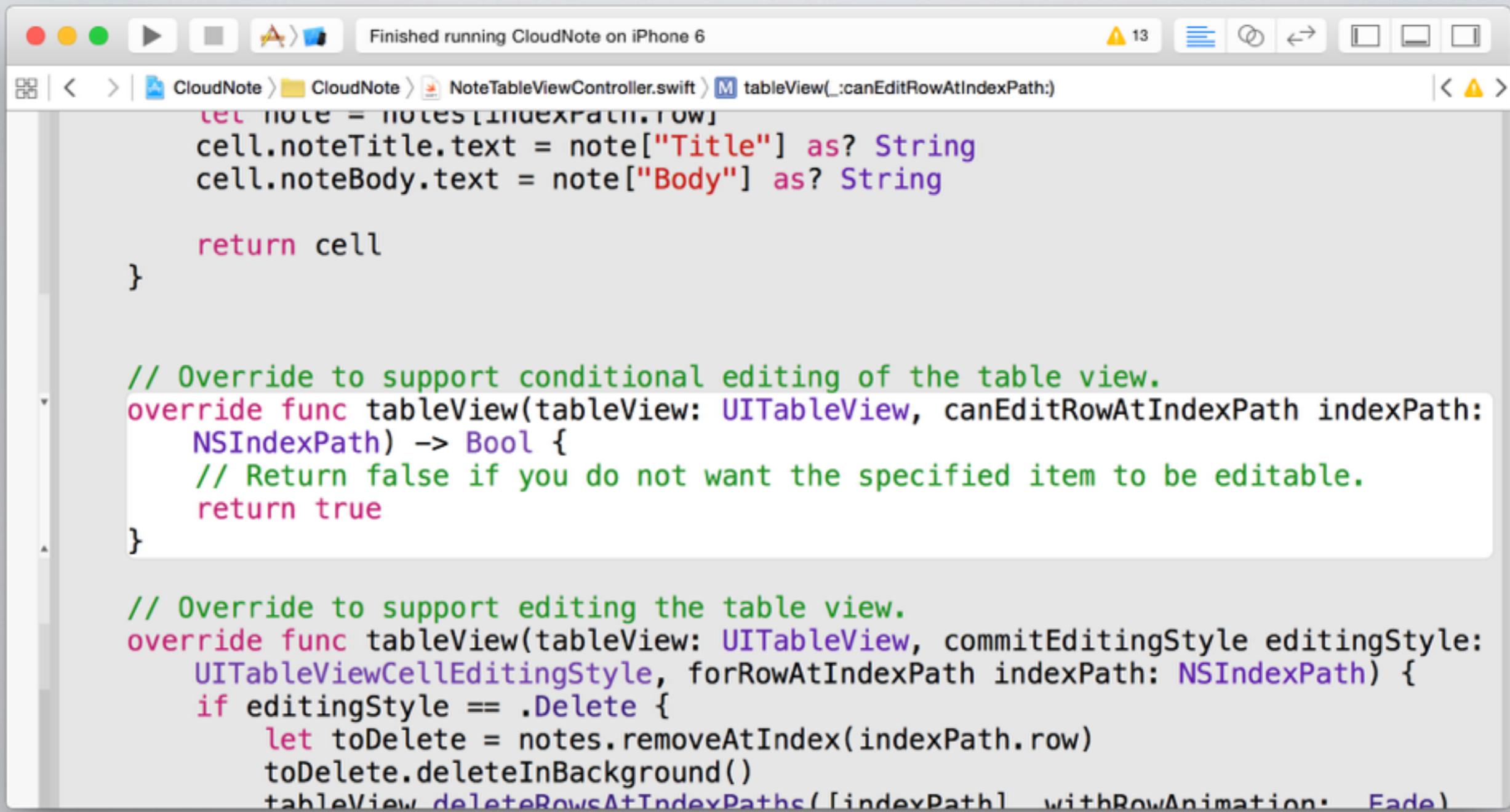
```
gameScore.deleteInBackground()
```

Objective-C **Swift**

If you want to run a callback when the delete is confirmed, you can use the `deleteInBackgroundWithBlock:` or `deleteInBackgroundWithTarget:selector:` methods. If you want to block the calling thread, you can use the `delete` method.

You can delete a single field from an object with the `removeObjectForKey` method:

DELETING



The screenshot shows the Xcode IDE interface with the following details:

- Toolbar:** Standard Xcode toolbar with icons for file, edit, run, stop, and build.
- Status Bar:** Shows "Finished running CloudNote on iPhone 6".
- Alert:** A yellow warning icon with "13" is visible in the top right corner.
- Project Navigator:** Shows the project structure: CloudNote > CloudNote > NoteTableViewCellController.swift.
- Code Editor:** Displays Swift code for a UITableView controller. The code includes methods for configuring table view cells, overriding editing behavior, and handling row deletion.

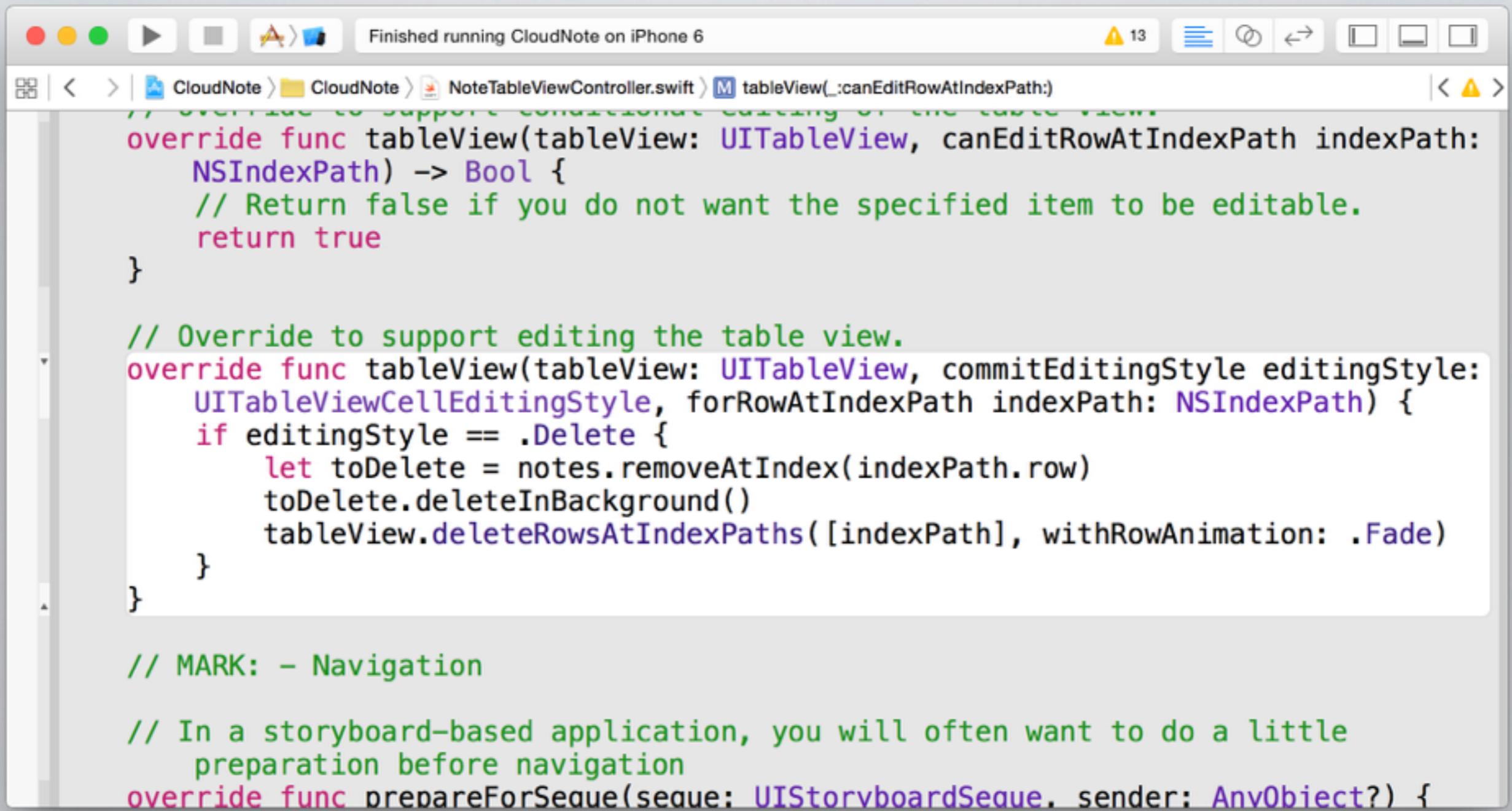
```
let note = notes[indexPath.row]
cell.noteTitle.text = note["Title"] as? String
cell.noteBody.text = note["Body"] as? String

return cell
}

// Override to support conditional editing of the table view.
override func tableView(tableView: UITableView, canEditRowAtIndexPath indexPath: NSIndexPath) -> Bool {
    // Return false if you do not want the specified item to be editable.
    return true
}

// Override to support editing the table view.
override func tableView(tableView: UITableView, commitEditingStyle editingStyle: UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {
    if editingStyle == .Delete {
        let toDelete = notes.removeAtIndex(indexPath.row)
        toDelete.deleteInBackground()
        tableView.deleteRowsAtIndexPaths([indexPath], withRowAnimation: .Fade)
    }
}
```

DELETING



The screenshot shows the Xcode IDE interface with the following details:

- Toolbar:** Standard Xcode toolbar with icons for running, stopping, and navigating.
- Status Bar:** Shows "Finished running CloudNote on iPhone 6".
- Alert:** A yellow warning icon with "13" alerts.
- File Navigator:** Shows the project structure: CloudNote > CloudNote > NoteTableViewController.swift.
- Text Editor:** Displays Swift code for a UITableViewController. The code includes methods for handling table view editing, specifically for deletion. It uses comments to explain the purpose of each section.

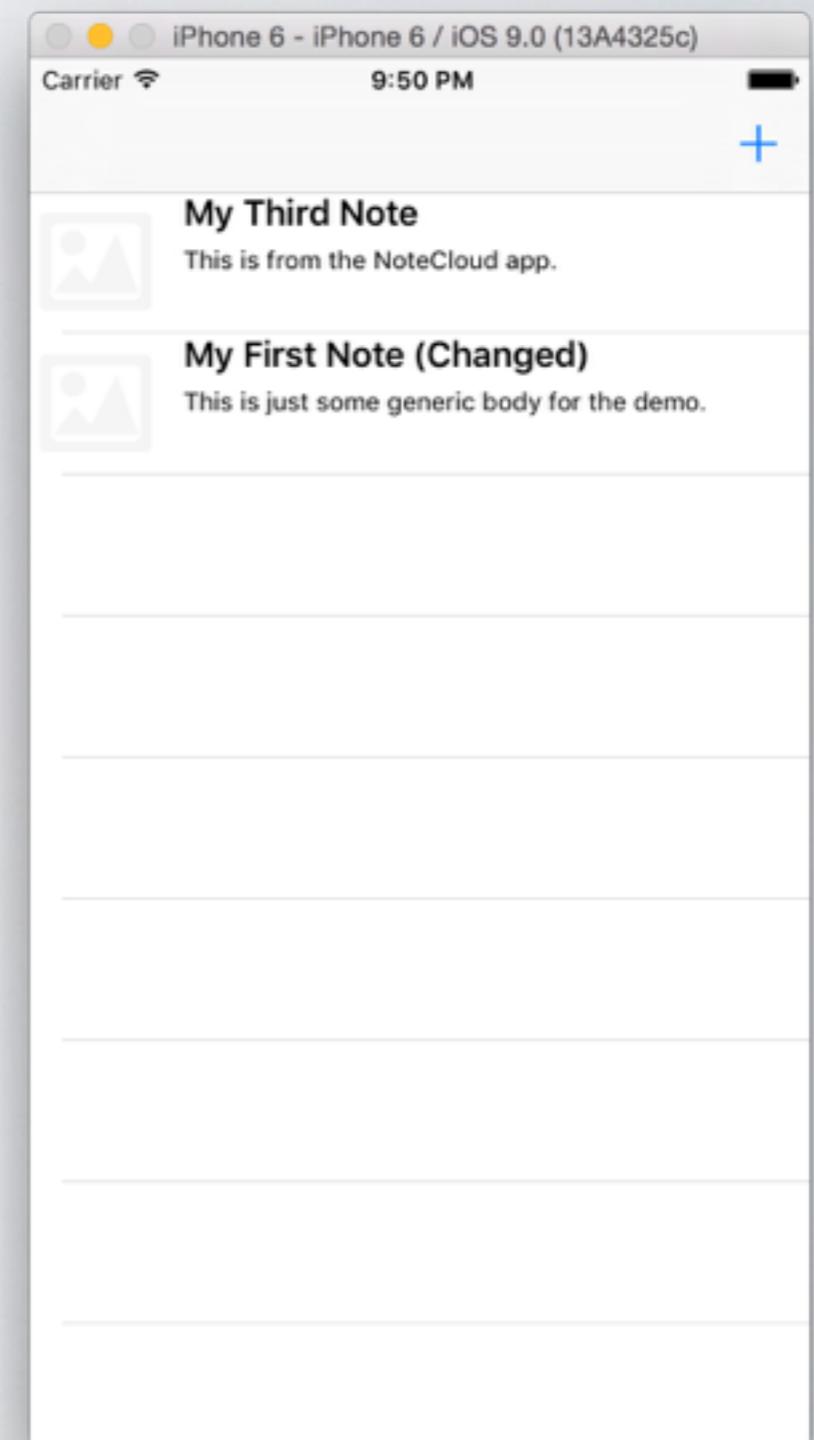
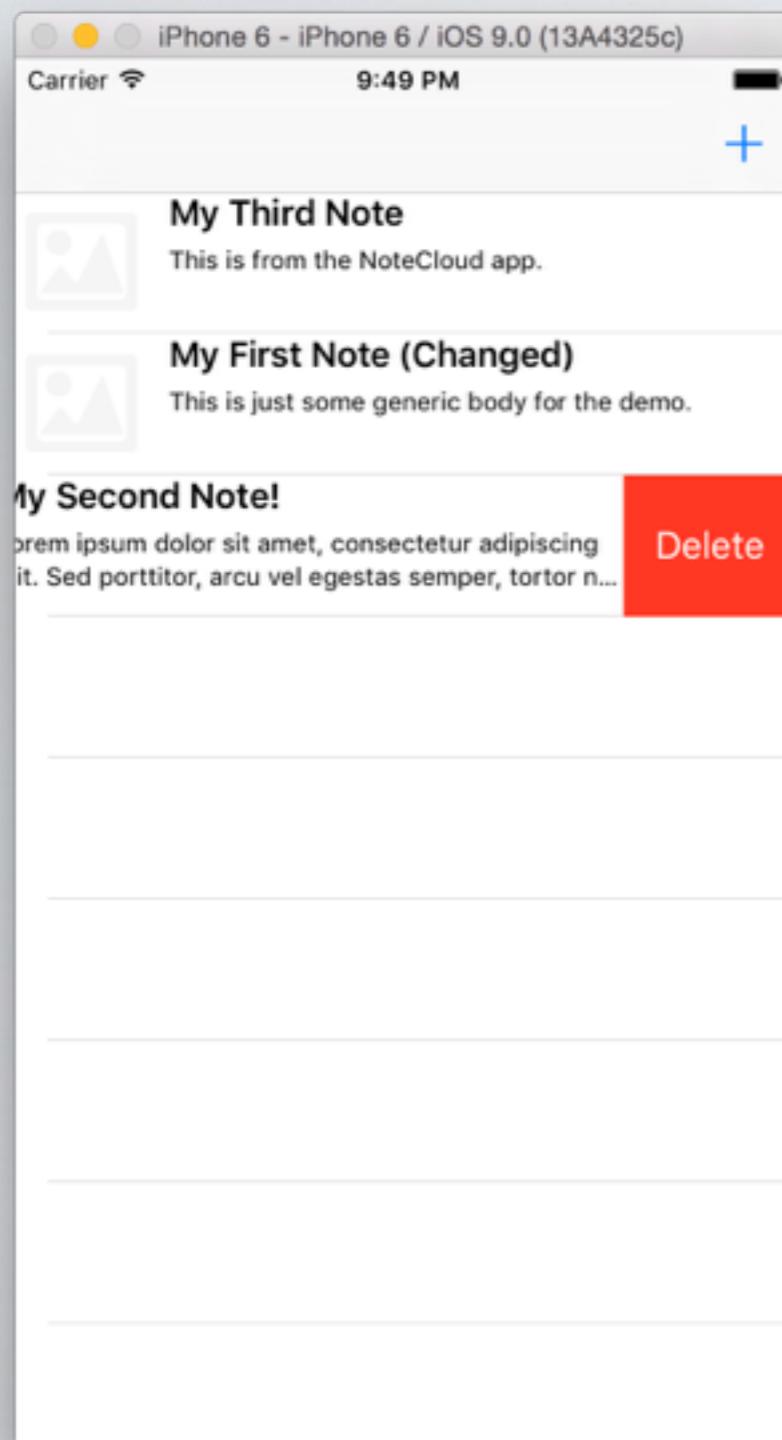
```
override func tableView(tableView: UITableView, canEditRowAtIndexPath indexPath: NSIndexPath) -> Bool {
    // Return false if you do not want the specified item to be editable.
    return true
}

// Override to support editing the table view.
override func tableView(tableView: UITableView, commitEditingStyle editingStyle: UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {
    if editingStyle == .Delete {
        let toDelete = notes.removeAtIndex(indexPath.row)
        toDelete.deleteInBackground()
        tableView.deleteRowsAtIndexPaths([indexPath], withRowAnimation: .Fade)
    }
}

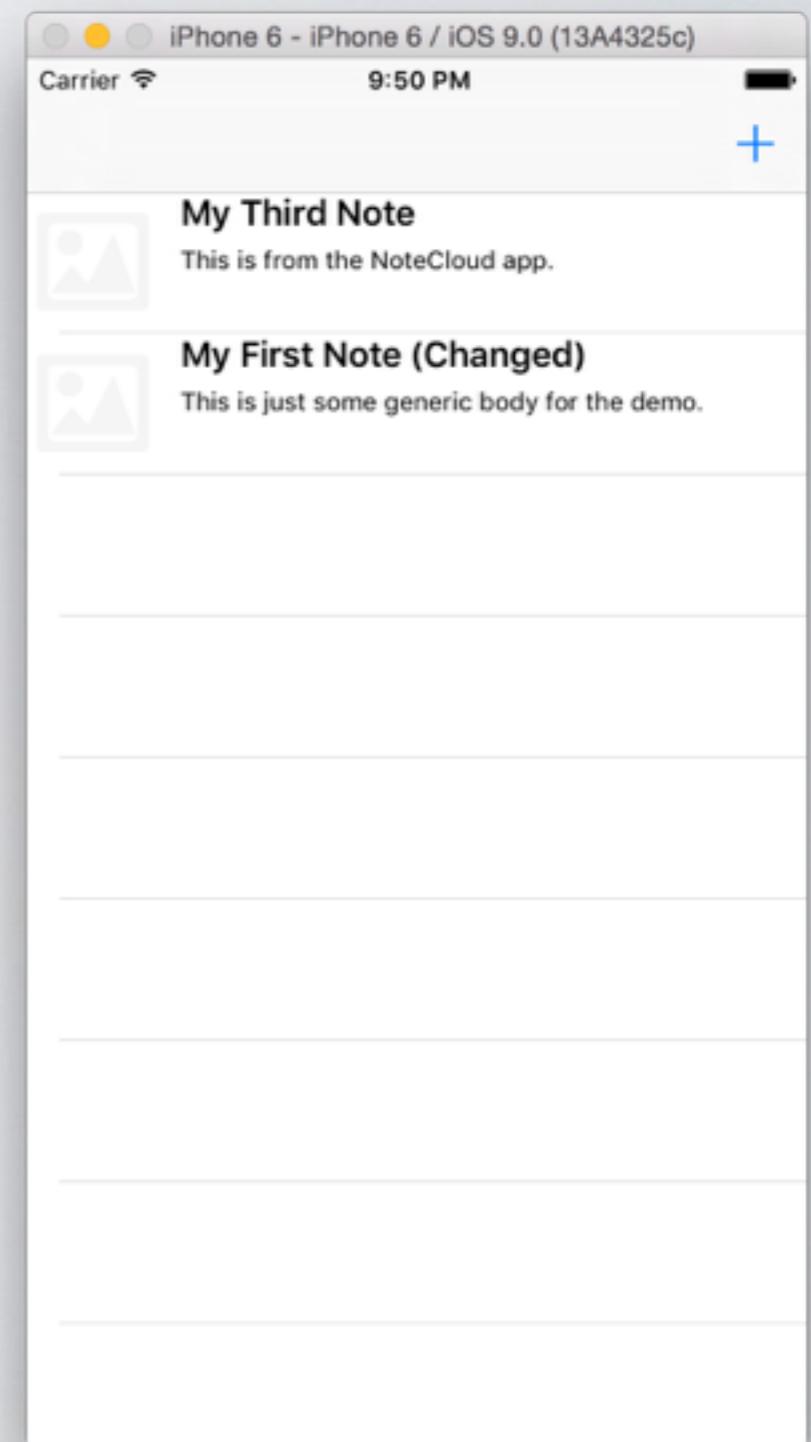
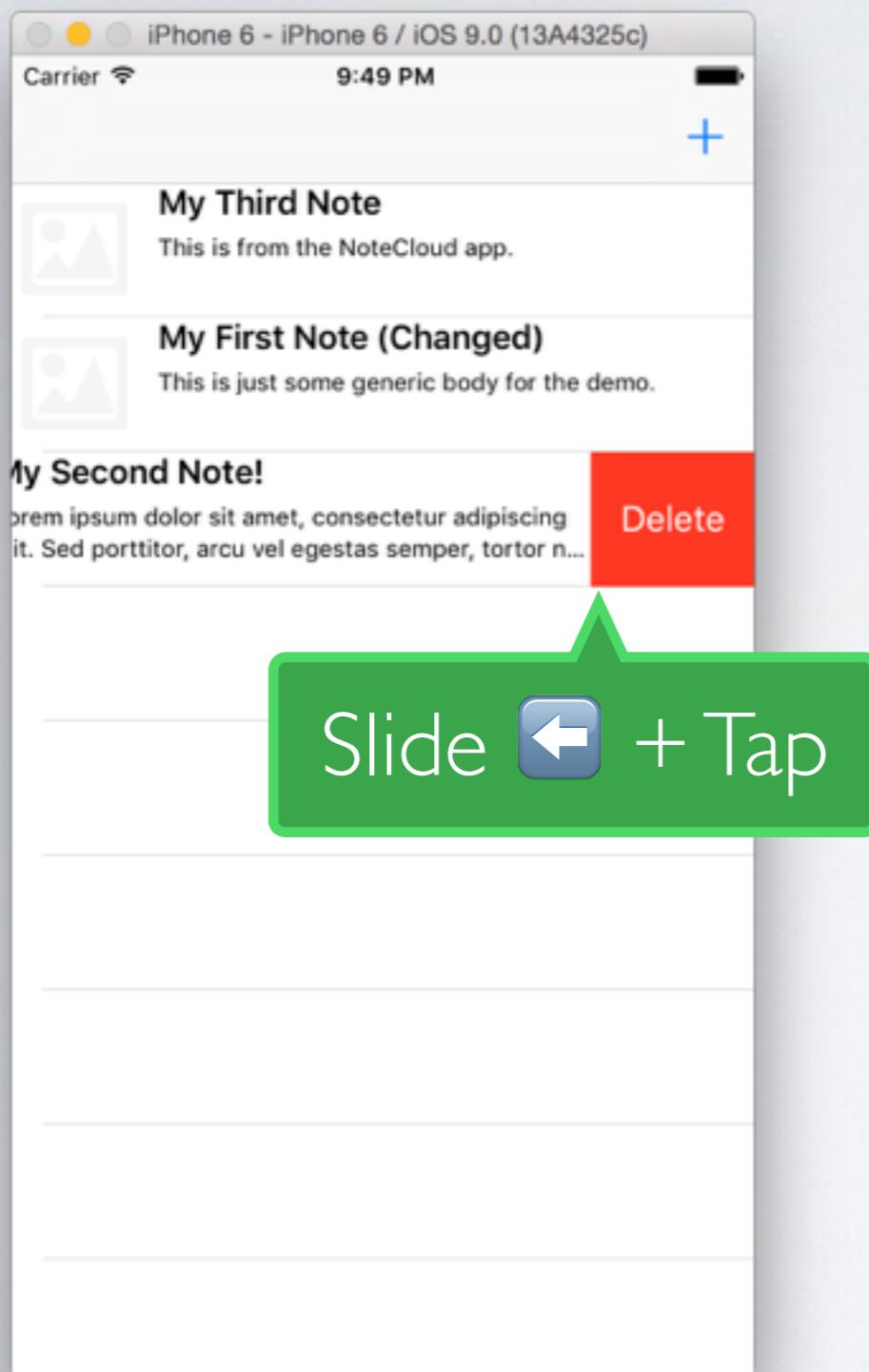
// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little
// preparation before navigation
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
```

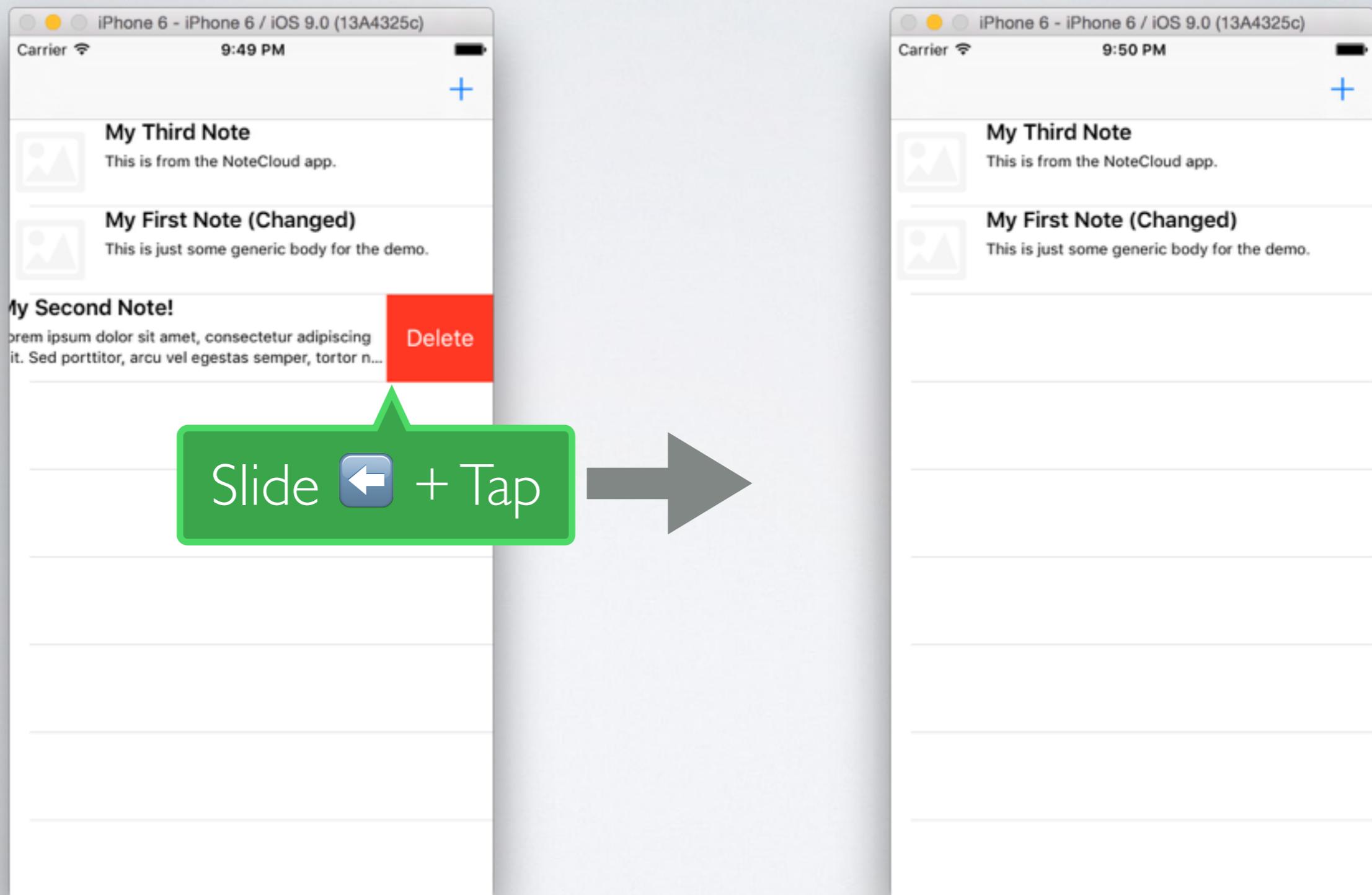
CURRENTVIEW



CURRENT VIEW



CURRENT VIEW



CHANGES ON PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a header bar with standard OS X window controls (red, yellow, green buttons), a back/forward button, a refresh icon, and a URL field showing www.parse.com/apps/cloudnote--4/collections. To the right of the URL are several icons: a share icon, a square icon, and a plus sign icon.

The main navigation bar includes the Parse logo, the app name "CloudNote" (with a dropdown menu showing "DEV"), and various core features: Core, Functions, Data, Security, and More. A user profile picture is also present.

The left sidebar contains links for "Data", "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config".

The central area is the "Data" tab, which displays a table of "Note" objects. The table has columns: objectID, Title, createdAt, and updatedAt. There are buttons for "+ Row", "- Row", "+ Col", "Security", "More", and a filter icon.

Note	objectID	Title	createdAt	updatedAt
	OzR79Cfyuo	My Third Note	Aug 11, 2015, 05:35	Aug 11, 2015, 05:35
	ug9aWR71Ud	My First Note (Changed)	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

Below the table are buttons for "+ Add Class" and "(+) Import".

At the bottom right, there are buttons for "20 rows/page" and a previous/next page navigation icon. The footer includes links for "Docs", "Billing", "Downloads", "Help", and "Status".

CHANGES ON PARSE

The screenshot shows the Parse.com dashboard interface. The top navigation bar includes standard OS X window controls, a lock icon, the URL www.parse.com/apps/cloudnote--4/collections, and various application icons. The main header displays the project name "CloudNote" and environment "DEV". Below the header, there's a toolbar with buttons for "+ Row", "- Row", "+ Col", "Security", "More", and a filter icon.

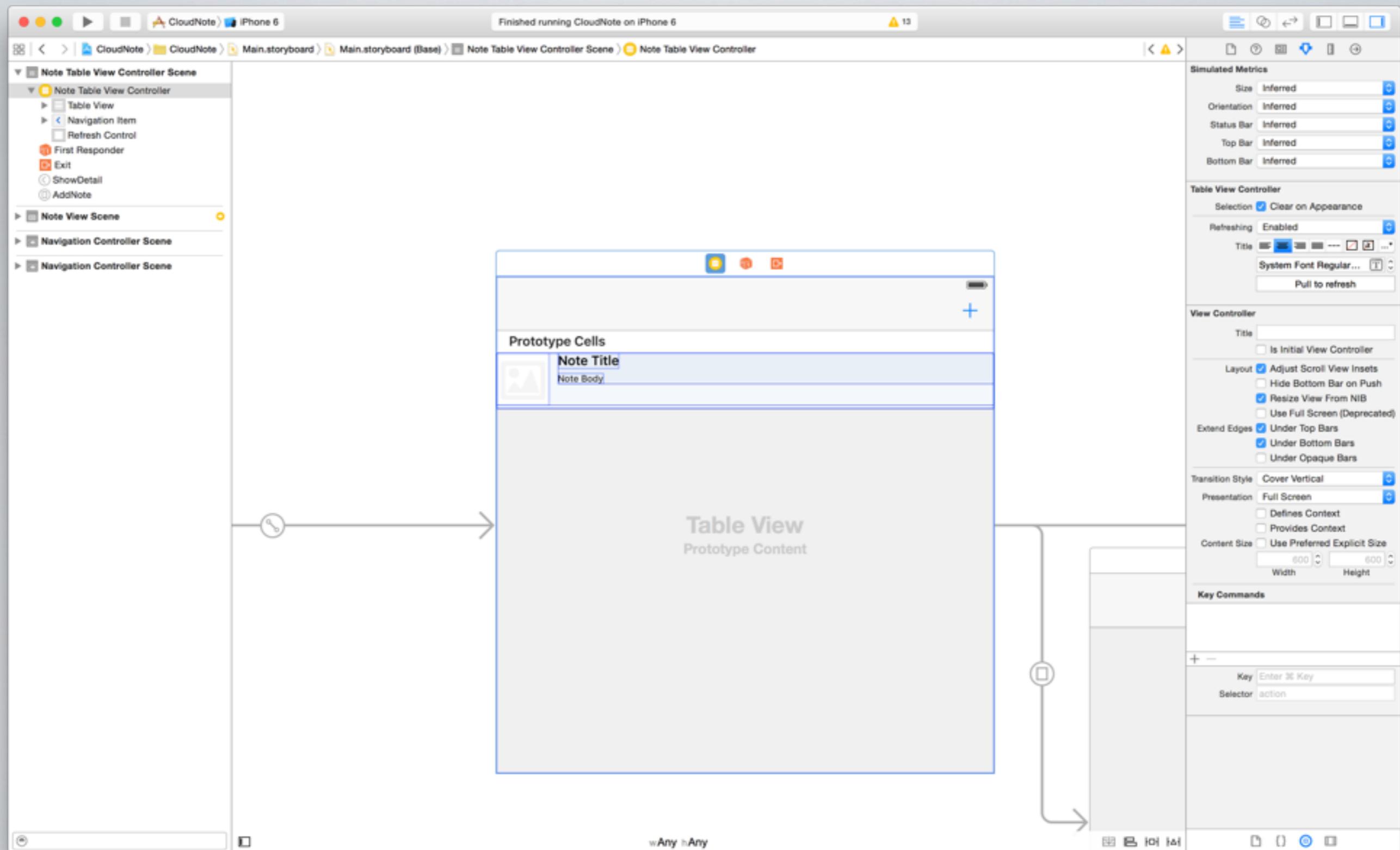
The left sidebar contains links for "Data", "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config". The "Data" link is currently selected, showing a table of data. The table has columns: Note, objectId, Title, createdAt, and updatedAt. There are two rows of data:

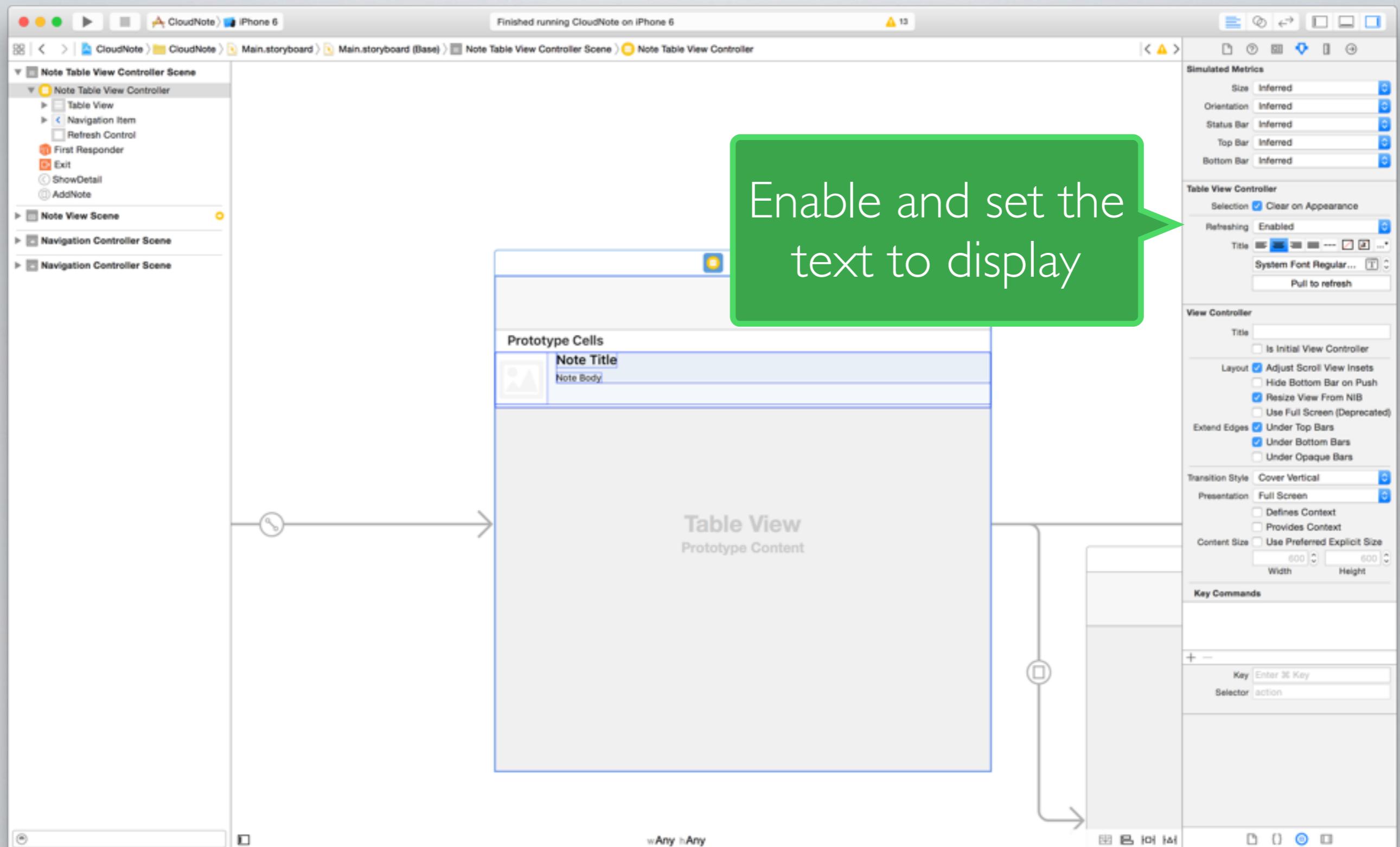
Note	objectId	Title	createdAt	updatedAt
2	OzR79Cfyuo	My Third Note	Aug 11, 2015, 05:35	Aug 11, 2015, 05:35
	ug9aWR71Ud	My First Note (Changed)	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

A green callout box with the text "Yup, note is deleted" points to the second row, indicating that the note has been removed from the collection.

At the bottom right, there are buttons for "20 rows/page" and a back arrow. The footer includes links for "Docs", "Billing", "Downloads", "Help", and "Stats".

TASK: PULL TO REFRESH





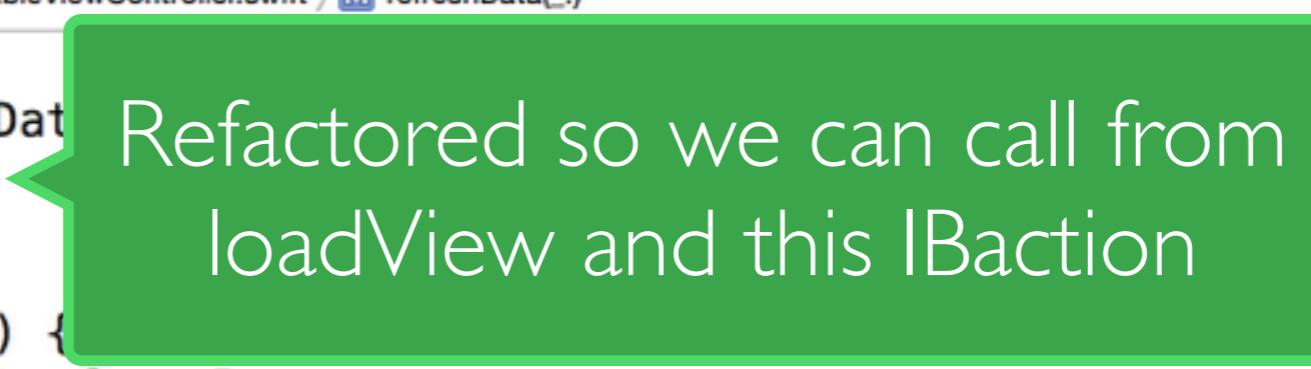
The screenshot shows the Xcode IDE interface with the following details:

- Top Bar:** Shows standard window controls (red, yellow, green circles, close, minimize, maximize) and a status message "Finished running CloudNote on iPhone 6".
- Toolbar:** Includes icons for file operations, search, and navigation.
- Document Outline:** Displays the project structure: CloudNote > CloudNote > NoteTableViewController.swift.
- Text Editor:** Contains the following Swift code:

```
// MARK: Refreshing
@IBAction func refreshData(sender: UIRefreshControl) {
    refreshFromParse()
}

func refreshFromParse() {
    // Load initial data from Parse
    // https://www.parse.com/docs/ios/guide#queries-basic-queries
    let query = PFQuery(className:"Note")
    query.addDescendingOrder("updatedAt")
    query.findObjectsInBackgroundWithBlock {
        (notes: [AnyObject]?, error: NSError?) -> Void in

        if error == nil {
            print("Successfully retrieved \(notes!.count) notes.")
            if let notes = notes as? [PFObject] {
                self.notes = notes
                self.tableView.reloadData()
                self.refreshControl?.endRefreshing()
            }
        } else {
    }
}
```



Finished running CloudNote on iPhone 6

CloudNote > CloudNote > NoteTableViewController.swift > refreshData(_:)

```
// MARK: Refreshing
@IBAction func refreshData(_ sender: AnyObject) {
    refreshFromParse()
}

func refreshFromParse() {
    // Load initial data from Parse
    // https://www.parse.com/docs/ios/guide#queries-basic-queries
    let query = PFQuery(className: "Note")
    query.addDescendingOrder("updatedAt")
    query.findObjectsInBackgroundWithBlock {
        (notes: [AnyObject]?, error: NSError?) -> Void in

        if error == nil {
            print("Successfully retrieved \(notes!.count) notes.")
            if let notes = notes as? [PFObject] {
                self.notes = notes
                self.tableView.reloadData()
                self.refreshControl?.endRefreshing()
            }
        } else {
            // Handle error
        }
    }
}
```

The screenshot shows the Xcode interface with the following details:

- Top Bar:** Shows standard Mac OS X window controls (red, yellow, green, close, minimize, maximize) and a status bar indicating "Finished running CloudNote on iPhone 6".
- Toolbar:** Standard Xcode toolbar with icons for file operations.
- Document Outline:** Shows the project structure: CloudNote > CloudNote > NoteTableViewController.swift.
- Code Editor:** Displays the following Swift code:

```
// MARK: Refreshing
@IBAction func refreshData(_ sender: AnyObject) {
    refreshFromParse()
}

func refreshFromParse() {
    // Load initial data from Parse
    // https://www.parse.com/docs/ios/guide#queries-basic-queries
    let query = PFQuery(className: "Note")
    query.addDescendingOrder("updatedAt")
    query.findObjectsInBackgroundWithBlock {
        (notes: [AnyObject]?, error: NSError?) -> Void in

        if error == nil {
            print("Successfully retrieved \(notes!.count) notes.")
            if let notes = notes as? [PFObject] {
                self.notes = notes
                self.tableView.reloadData()
                self.refreshControl?.endRefreshing()
            }
        } else {
            // Error handling logic
        }
    }
}
```

Annotations:

- A green callout bubble points to the `refreshFromParse()` method with the text: "Refactored so we can call from loadView and this IBAction".
- A red callout bubble points to the `endRefreshing()` call with the text: "Remember to call endRefreshing()".

CHANGES ON PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a header bar with standard OS X window controls (red, yellow, green buttons), a back/forward button, a refresh icon, and a URL bar displaying "www.parse.com/apps/cloudnote--4/collections". To the right of the URL are several icons: a file, a plus sign, a user profile, and a settings gear.

The main navigation bar includes the Parse logo, the app name "CloudNote" (with a dropdown menu showing "DEV"), and tabs for "Core", "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config".

The left sidebar has a "Data" section with a table showing three rows of data for the "Note" class. The columns are: objectID (String), Title (String), createdAt (Date), and updatedAt (Date). The rows are:

Note	objectID	Title	createdAt	updatedAt
	vU8CbfvakQ	Just added	Aug 13, 2015, 05:43	Aug 13, 2015, 05:43
	OzR79Cfyuo	My Third Note	Aug 11, 2015, 05:35	Aug 11, 2015, 05:35
	ug9aWR71Ud	My First Note (Changed)	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

Below the table are buttons for "+ Add Class" and "(+) Import".

The bottom right corner of the dashboard shows pagination controls: "20" (rows/page) and a left arrow icon.

At the very bottom, there are links for "Docs", "Billing", "Downloads", "Help", and "Stats".

CHANGES ON PARSE

The screenshot shows the Parse.com dashboard interface. At the top, there's a header with browser controls, a lock icon, the URL www.parse.com/apps/cloudnote--4/collections, and various application and system icons.

The main navigation bar includes the Parse logo, the app name "CloudNote" (with a dropdown menu), a "DEV" switch, and tabs for "Core", "Cloud Code", "Webhooks", "Jobs", "Logs", and "Config".

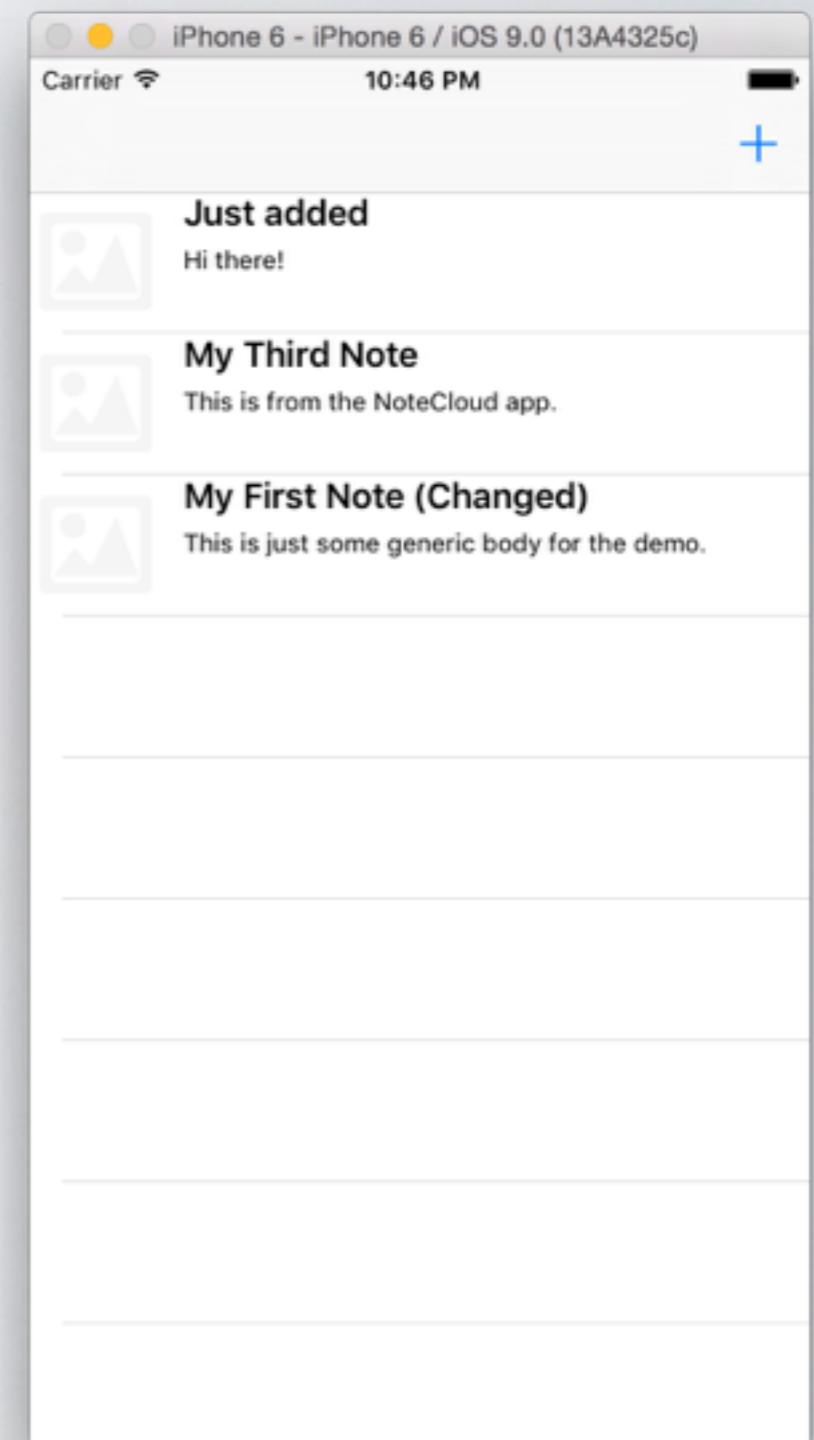
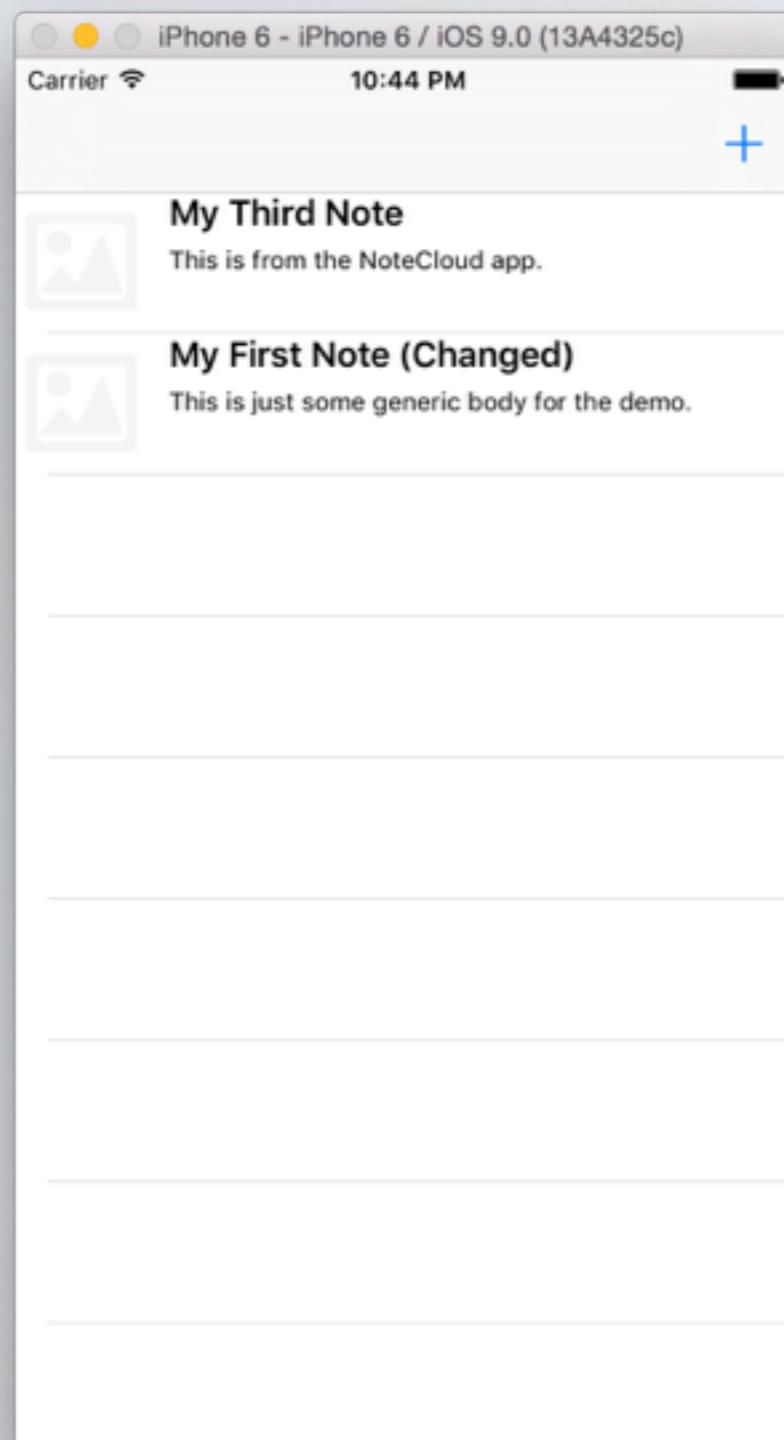
The left sidebar has a "Data" section with a table showing three rows of data for the "Note" class. The columns are "objectId", "Title", "createdAt", and "updatedAt". The first row has the title "Just added". A green callout box with the text "Add a new note" is overlaid on the table area.

The bottom right corner of the dashboard shows pagination controls: "20" rows/page and a back/forward navigation icon.

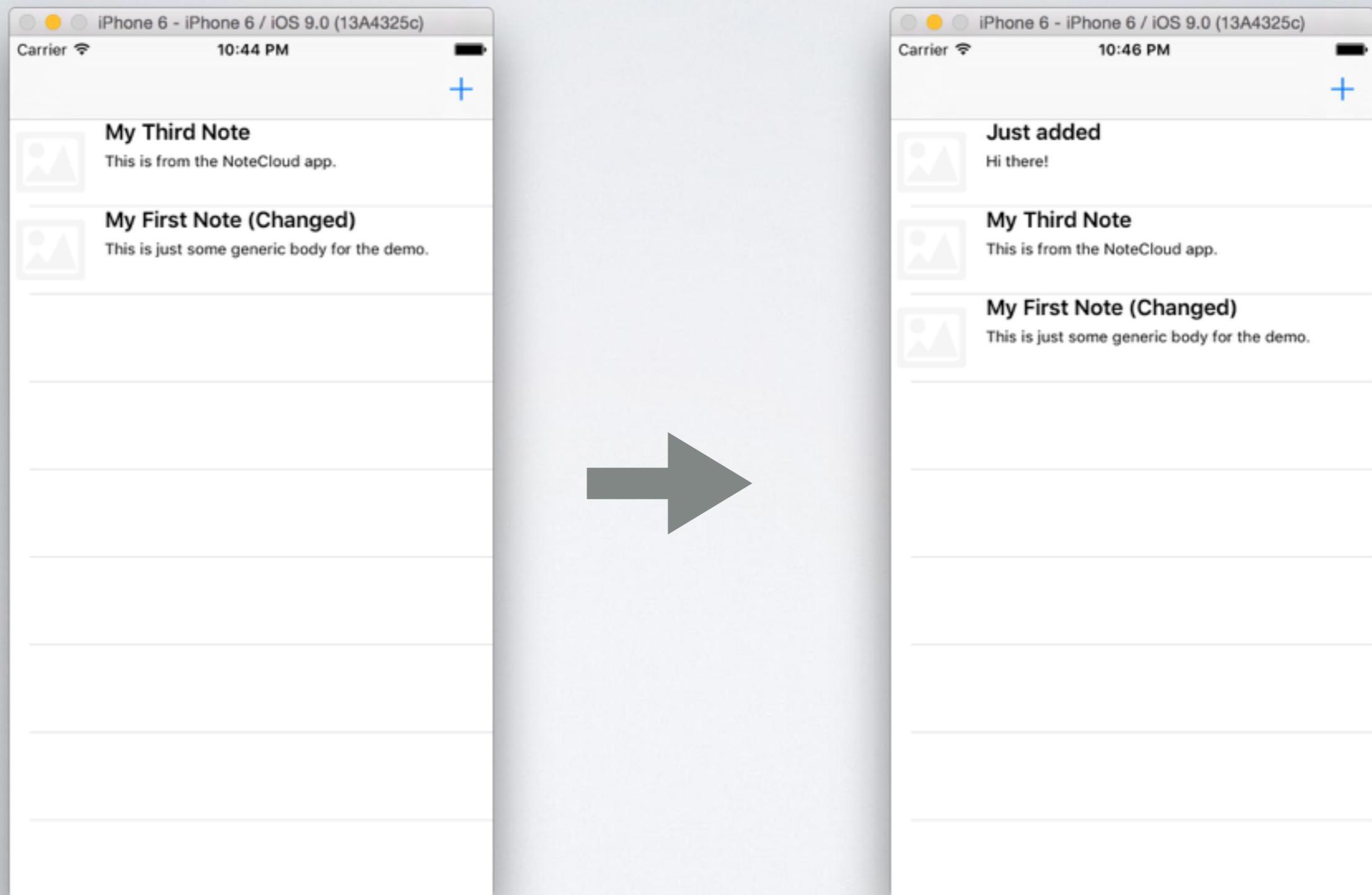
	objectId	Title	createdAt	updatedAt
<input type="checkbox"/>	vU8CbfvakQ	Just added	Aug 13, 2015, 05:43	Aug 13, 2015, 05:43
<input type="checkbox"/>	OzR790C	My first note	Aug 11, 2015, 05:35	Aug 11, 2015, 05:35
<input type="checkbox"/>	ug9aWP	Test note	Aug 09, 2015, 04:12	Aug 10, 2015, 04:12

Bottom navigation links include "Docs", "Billing", "Downloads", "Help", and "Stats".

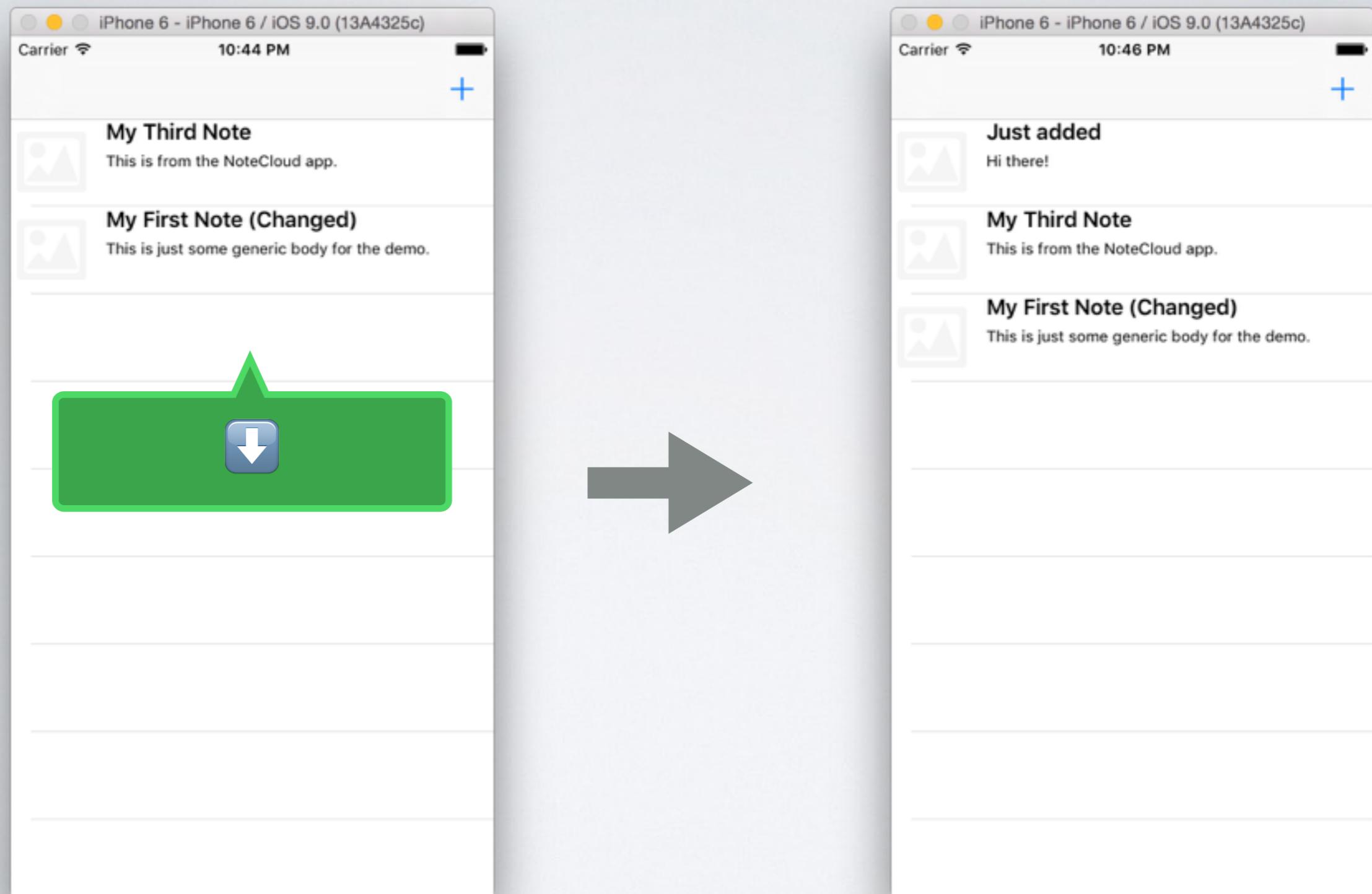
CURRENTVIEW



CURRENTVIEW

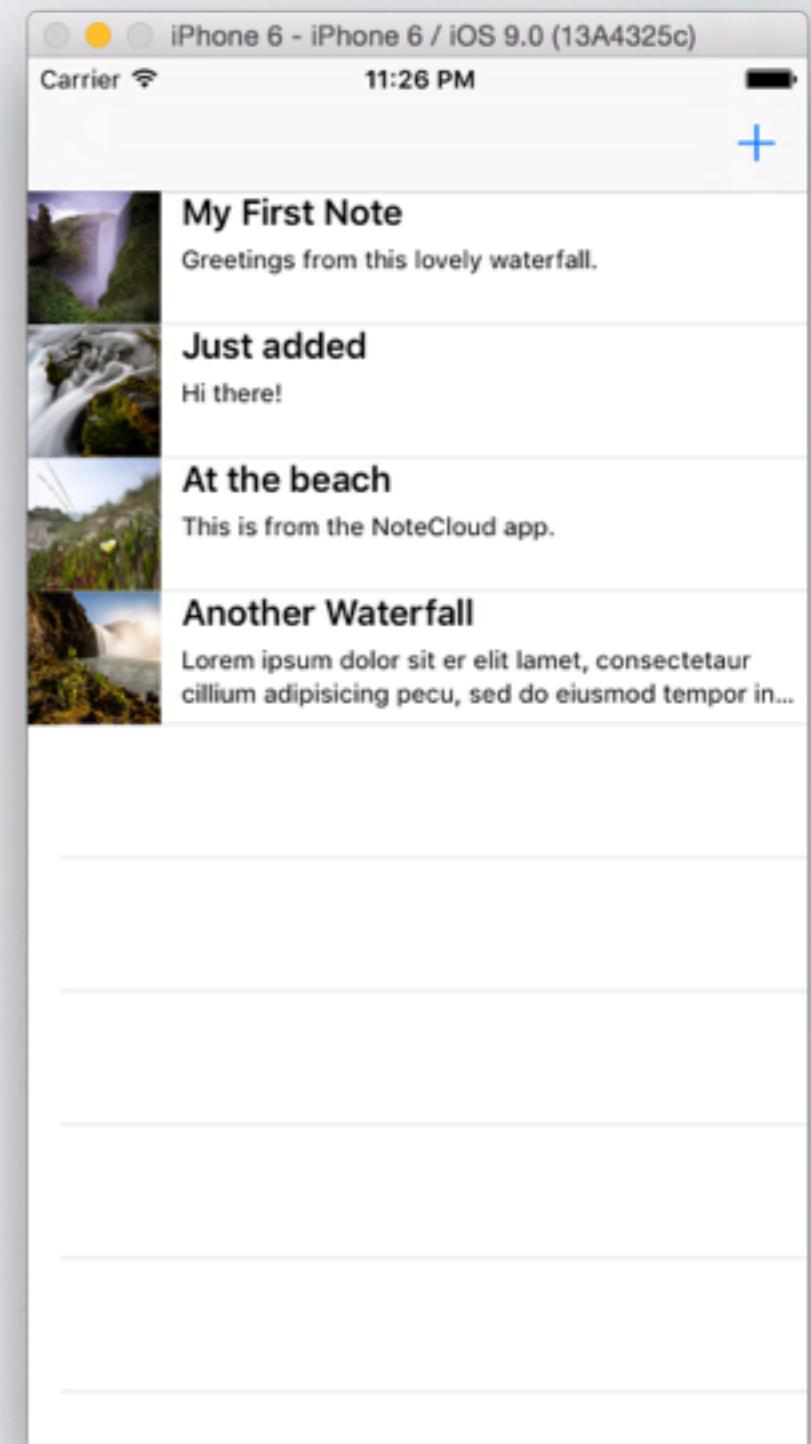


CURRENT VIEW



EXERCISE

- Make it possible to save the photo to PARSE



NEXT TIME

- Animation - lots of fun stuff