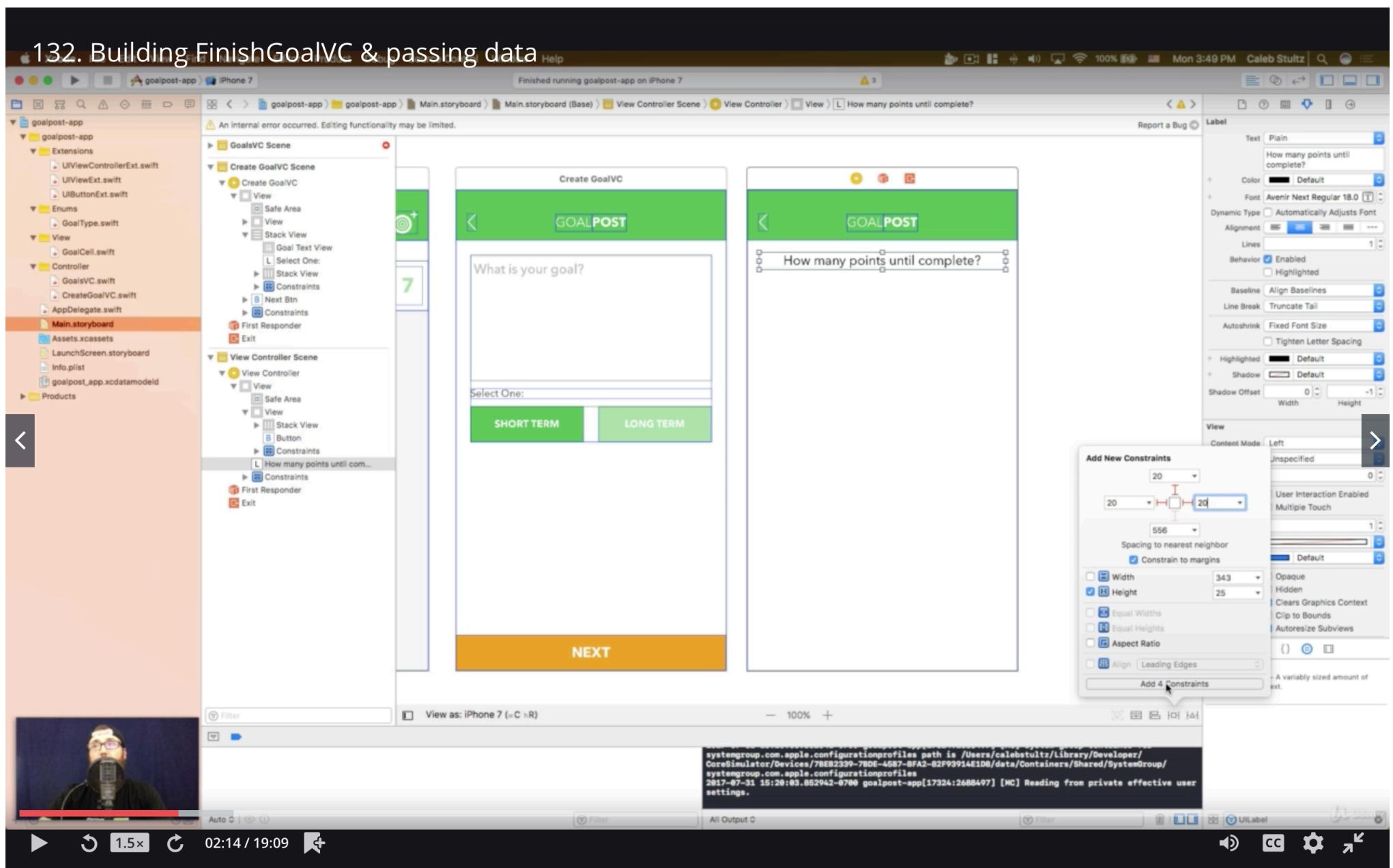


## 132. Building FinishGoalVC & passing data



## 132. Building FinishGoalVC & passing data

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "goalpost-app".
- File Navigator:** Shows files like "Main.storyboard", "FinishGoalVC.swift", and "GoalsVC.swift".
- Editor:** Displays the code for "FinishGoalVC.swift".
- Storyboard Preview:** Shows a green "GOAL POST" screen with a text field containing "0" and a "CREATE GOAL" button.
- Action Connection Inspector:** A modal window is open, showing a connection from the "CREATE GOAL" button to an action named "createGoalBtrnWasPressed".
- Code Editor:** The code for "FinishGoalVC.swift" includes comments and an implementation of the `viewDidLoad` method.
- Output Navigator:** Shows system logs related to configuration profiles.
- Bottom Bar:** Includes playback controls, a video preview of the speaker, and a timestamp of 05:24 / 19:09.

```
//
// FinishGoalVC.swift
// goalpost-app
//
// Created by Caleb Stultz on 7/31/17.
// Copyright © 2017 Caleb Stultz. All rights reserved.

import UIKit

class FinishGoalVC: UIViewController {

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

## 132. Building FinishGoalVC & passing data

The screenshot shows the Xcode interface with two main panes. On the left, the storyboard editor displays 'Main.storyboard' with a scene for 'Finish GoalVC'. This scene contains a stack view with a text field labeled 'How many points until complete?' and a placeholder '0'. Below the stack view is a large orange button labeled 'CREATE GOAL'. On the right, the code editor shows 'FinishGoalVC.swift' with the following Swift code:

```
// goalpost-app
// Created by Caleb Stultz on 7/31/17.
// Copyright © 2017 Caleb Stultz. All rights reserved.

// Variables of next VC to pass data.

import UIKit

class FinishGoalVC: UIViewController {

    @IBOutlet weak var createGoalBtn: UIButton!
    @IBOutlet weak var pointsTextField: UITextField!

    var goalDescription: String!
    var goalType: GoalType!

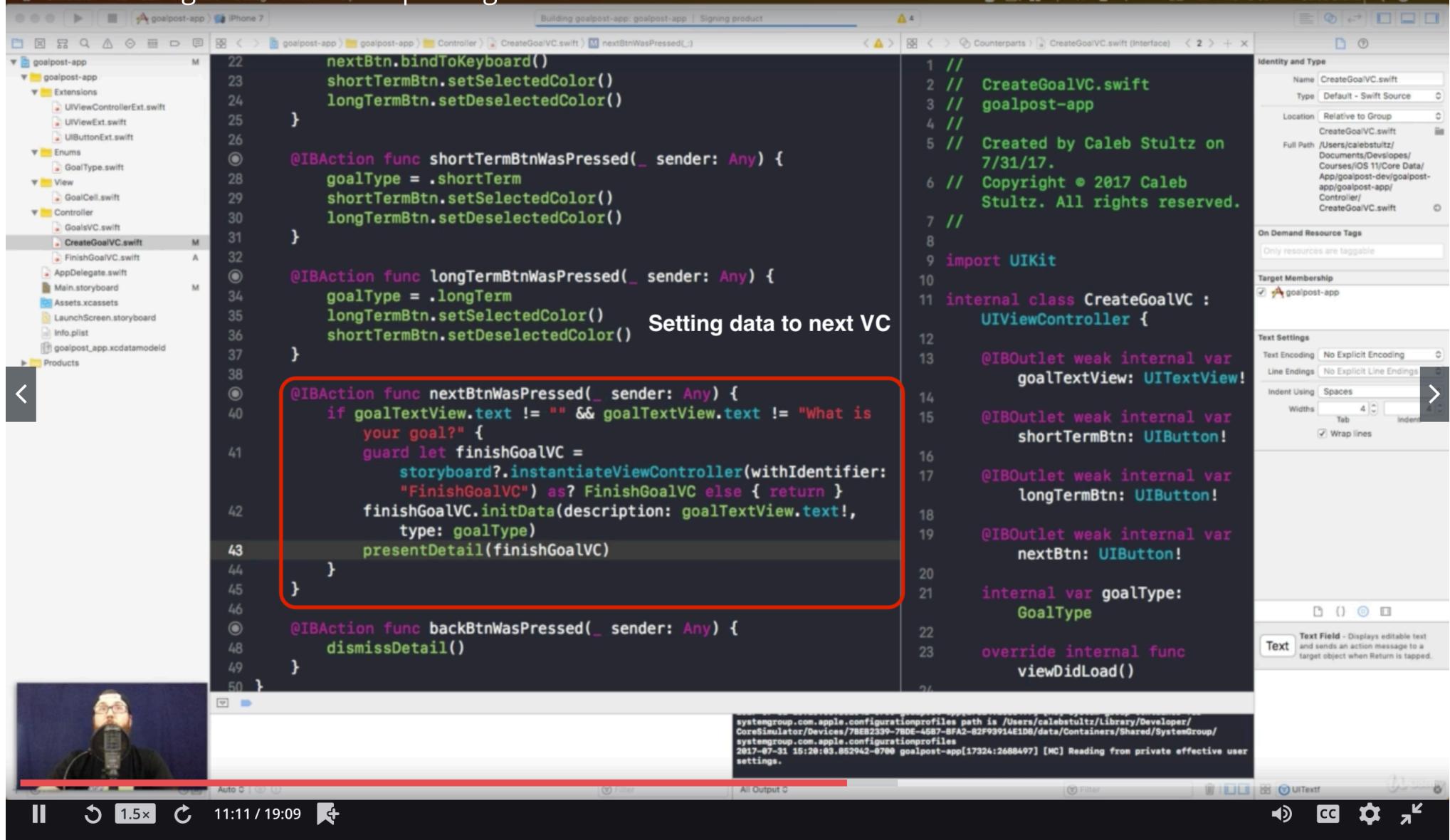
    func initData(description: String, type: GoalType) {
        self.goalDescription = description
        self.goalType = type
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        createGoalBtn.bindToKeyboard()
    }

    @IBAction func createGoalBtnWasPressed(_ sender: Any) {
        // Pass data into Core Data Goal Model
    }
}
```

A red box highlights the variable declarations at the top and the `initData` function. A second red box highlights the entire `initData` function. The bottom right corner of the code editor has a tooltip for the `Text` field: "Text Field - Displays editable text and sends an action message to a target object when Return is tapped." The bottom status bar shows the video feed of the speaker, the time 08:03 / 19:09, and playback controls.

## 132. Building FinishGoalVC & passing data



The screenshot shows the Xcode interface with the following details:

- Project Structure:** The left sidebar shows the project structure for "goalpost-app".
- Code Editor:** The main editor displays the "CreateGoalVC.swift" file. A red box highlights the following code block:

```
        @IBAction func nextBtnWasPressed(_ sender: Any) {
            if goalTextView.text != "" && goalTextView.text != "What is
                your goal?" {
                guard let finishGoalVC =
                    storyboard?.instantiateViewController(withIdentifier:
                        "FinishGoalVC") as? FinishGoalVC else { return }
                finishGoalVC.initData(description: goalTextView.text!,
                                      type: goalType)
                presentDetail(finishGoalVC)
            }
        }
```

- Text View:** The right side of the screen contains a text view with the following content:

```
// CreateGoalVC.swift
// goalpost-app
// Created by Caleb Stultz on
// 7/31/17.
// Copyright © 2017 Caleb
// Stultz. All rights reserved.

import UIKit
internal class CreateGoalVC : UIViewController {

    @IBOutlet weak internal var
    goalTextView: UITextView!

    @IBOutlet weak internal var
    shortTermBtn: UIButton!

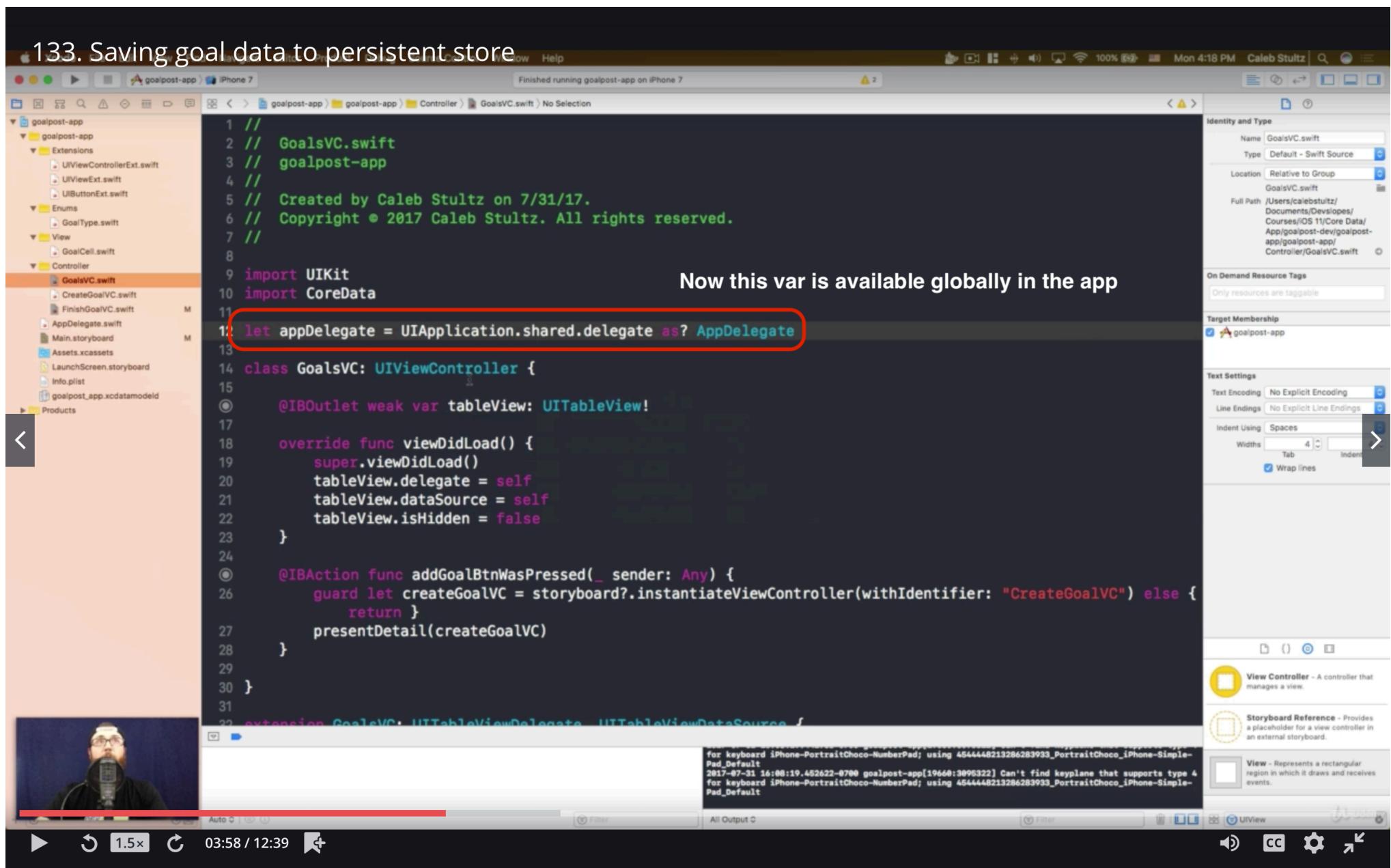
    @IBOutlet weak internal var
    longTermBtn: UIButton!

    @IBOutlet weak internal var
    nextBtn: UIButton!

    internal var goalType:
    GoalType

    override internal func
    viewDidLoad()
```
- Output Log:** The bottom log window shows system logs related to configuration profiles.
- Video Player:** A video player in the bottom-left corner shows a man speaking.

### 133. Saving goal data to persistent store



The screenshot shows the Xcode interface with the code editor open to the `GoalsVC.swift` file. A red box highlights the line of code where `appDelegate` is assigned:

```
1 //  
2 //  GoalsVC.swift  
3 //  goalpost-app  
4 //  
5 //  Created by Caleb Stultz on 7/31/17.  
6 //  Copyright © 2017 Caleb Stultz. All rights reserved.  
7 //  
8  
9 import UIKit  
10 import CoreData  
11  
12 let appDelegate = UIApplication.shared.delegate as? AppDelegate  
13  
14 class GoalsVC: UIViewController {  
15     ...  
16     @IBOutlet weak var tableView: UITableView!  
17  
18     override func viewDidLoad() {  
19         super.viewDidLoad()  
20         tableView.delegate = self  
21         tableView.dataSource = self  
22         tableView.isHidden = false  
23     }  
24  
25     @IBAction func addGoalBtnWasPressed(_ sender: Any) {  
26         guard let createGoalVC = storyboard?.instantiateViewController(withIdentifier: "CreateGoalVC") else {  
27             return }  
28         presentDetail(createGoalVC)  
29     }  
30 }  
31  
32 extension GoalsVC: UITableViewDelegate, UITableViewDataSource {
```

A callout bubble points to the `appDelegate` variable with the text: "Now this var is available globally in the app".

The right side of the Xcode interface shows the `identity and Type` inspector for the `GoalsVC.swift` file, and the bottom right shows the `View Controller`, `Storyboard Reference`, and `View` documentation cards.

### 133. Saving goal data to persistent store

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "goalpost-app".
- Editor:** Displays the code for `FinishGoalVC.swift`. A red box highlights the `save(completion:)` function.
- Text Editor Content:**

```
33 }
34
35 @IBAction func backBtnWasPressed(_ sender: Any) {
36     dismissDetail()
37 }
38
39 func save(completion: (_ finished: Bool) -> ()) {
40     guard let managedContext = appDelegate?.persistentContainer.viewContext else { return }
41     let goal = Goal(context: managedContext)
42
43     goal.goalDescription = goalDescription
44     goal.goalType = goalType.rawValue
45     goal.goalCompletionValue = Int32(pointsTextField.text!)!
46     goal.goalProgress = Int32(0)
47
48     do {
49         try managedContext.save()
50     } catch {
51         debugPrint("Could not save: \(error.localizedDescription)")
52     }
53 }
```
- Identity and Type:** Shows the file is a Swift source file named `FinishGoalVC.swift`.
- On Demand Resource Tags:** Shows "Only resources are taggable".
- Target Membership:** Shows the target is `goalpost-app`.
- Text Settings:** Shows text encoding as "No Explicit Encoding" and line endings as "No Explicit Line Endings".
- Output Window:** Shows a warning message about a missing keychain type 4 for the keyboard.
- Bottom Bar:** Shows playback controls, a progress bar at 09:29 / 12:39, and other standard OS X application controls.

**Saving Goal data to core data**

### 133. Saving goal data to persistent store

The screenshot shows the Xcode interface with the following details:

- Project Structure:** The project is named "goalpost-app". The file "FinishGoalVC.swift" is open in the editor.
- Editor Content:** The code implements a view controller with methods for saving goals to a persistent store.
- Red Box Selection:** A red box highlights the following code block:

```
    @IBAction func createGoalBtnWasPressed(_ sender: Any) {
        if pointsTextField.text != "" {
            self.save { (complete) in
                if complete {
                    dismiss(animated: true, completion: nil)
                }
            }
        }
    }
```

- Output Navigator:** Shows system logs indicating keyboard configuration issues:

```
2017-07-31 16:08:09.866258-0700 goalpost-app[19660:3095322] [MC] Reading from private effective user settings
2017-07-31 16:08:19.441938-0700 goalpost-app[19660:3095322] Can't find keylane that supports type 4
for keyboard iPhone-PortraitChoco-NumberPad; using 4644448213286283933_PortraitChoco_iPhone-Simple-
Pad_Default
2017-07-31 16:08:19.482622-0700 goalpost-app[19660:3095322] Can't find keylane that supports type 4
for keyboard iPhone-PortraitChoco-NumberPad; using 4644448213286283933_PortraitChoco_iPhone-Simple-
Pad_Default
```

- Right Sidebar:** Shows the file's identity and type, target membership, and text settings.
- Bottom Bar:** Includes a video camera icon, a play/pause button, a refresh button, a zoom level indicator (1x), a timestamp (11:25 / 12:39), and other standard OS X interface elements.

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

goalpost-app iPhone 7 Finished running goalpost-app on iPhone 7 Mon 5:16 PM Caleb Stultz

goalpost-app  
Extensions  
Enums  
View  
Controller  
goalpost\_app.xcdatamodeld  
Products

```
1 //  
2 //  GoalsVC.swift  
3 //  goalpost-app  
4 //  
5 //  Created by Caleb Stultz on 7/31/17.  
6 //  Copyright © 2017 Caleb Stultz. All rights reserved.  
7 //  
8  
9 import UIKit  
10 import CoreData  
11  
12 let appDelegate = UIApplication.shared.delegate as? AppDelegate  
13  
14 class GoalsVC: UIViewController {  
15  
16     @IBOutlet weak var tableView: UITableView!  
17  
18     var goals: [Goal]  
19  
20     override func viewDidLoad() {  
21         super.viewDidLoad()  
22         tableView.delegate = self  
23         tableView.dataSource = self  
24         tableView.isHidden = false  
25     }  
26  
27     @IBAction func addGoalBtnWasPressed(_ sender: Any) {  
28         guard let createGoalVC = storyboard?.instantiateViewController(withIdentifier: "CreateGoalVC") else {  
29             return }  
30         presentDetail(createGoalVC)  
}
```

identity and Type  
Name: GoalsVC.swift  
Type: Default - Swift Source  
Location: Relative to Group  
Full Path: /Users/calebstultz/Documents/Devs/ios/Courses/iOS 11/Core Data/App/goalpost-dev/goalpost-app/goalpost-app/Controller/GoalsVC.swift  
On Demand Resource Tags  
Only resources are taggable  
Target Membership  
goalpost-app  
Text Settings  
Text Encoding: No Explicit Encoding  
Line Endings: No Explicit Line Endings  
Indent Using: Spaces  
Widths: Tab 4 Indent 4  
Wrap lines  
View Controller - A controller that manages a view.  
Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.  
View - Represents a rectangular region in which it draws and receives events.

Auto ⌂ Filter All Output ⌂ Filter ⌂ UIView



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

goalpost-app | Build goalpost-app: Failed | Today at 5:22 PM 100% 1 Mon 5:22 PM Caleb Stultz

goalpost-app

Extensions

Enums

View

Controller

goalpost\_app.xcdatamodeld

Products

goalpost-app

UIViewControllerExt.swift

UIViewExt.swift

UIButtonExt.swift

GoalType.swift

GoalCell.swift

GoalsVC.swift

CreateGoalVC.swift

FinishGoalVC.swift

AppDelegate.swift

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

goalpost-app.xcdatamodeld

goalpost-app

54 }

55

56 func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

57 guard let cell = tableView.dequeueReusableCell(withIdentifier: "goalCell") as? GoalCell else { return

58 UITableViewCell() }

59 let goal = goals[indexPath.row]

60 cell.configureCell(description: goal.goalDescription!, type: goal.goalType, goalProgressAmount:

61 goal.goalProgress)

62 return cell

63

64 extension GoalsVC {

65 func fetch(completion: (\_ complete: Bool) -> ()) {

66 guard let managedContext = appDelegate?.persistentContainer.viewContext else { return }

67

68 let fetchRequest = NSFetchedResultsController<Goal>(entityName: "Goal")

69

70 do {

71 goals = try managedContext.fetch(fetchRequest)

72 print("Successfully fetched data.")

73 completion(true)

74 } catch {

75 debugPrint("Could not fetch: \(error.localizedDescription)")

76 completion(false)

77 }

78 }

79 }

80

81

82

Build Failed

2017-07-31 16:38:50.004133-0700 goalpost-app[21140:3389565] Can't find keyplane that supports type 4 for keyboard iPhone-PortraitChoco-NumberPad; using 4644448213286283933\_PortraitChoco\_iPhone-SimplePad\_Default

2017-07-31 16:38:50.004133-0700 goalpost-app[21140:3389565] Can't find keylane that supports type 4 for keyboard iPhone-PortraitChoco-NumberPad; using 4644448213286283933\_PortraitChoco\_iPhone-SimplePad\_Default

Successfully saved data.

identity and Type

Name: GoalsVC.swift

Type: Default - Swift Source

Location: Relative to Group

Full Path: /Users/caiebstultz/Documents/Devs/ios/Courses/iOS 11/Core Data/App/goalpost-dev/goalpost-app/goalpost-app/Controller/GoalsVC.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

goalpost-app

Text Settings

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: Tab 4 Indent 4

Wrap lines

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

View - Represents a rectangular region in which it draws and receives events.



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

goalpost-app iPhone 7 Finished running goalpost-app on iPhone 7

```
8
9 import UIKit
10 import CoreData
11
12 let appDelegate = UIApplication.shared.delegate as? AppDelegate
13
14 class GoalsVC: UIViewController {
15
16     @IBOutlet weak var tableView: UITableView!
17
18     var goals: [Goal] = []
19
20     override func viewDidLoad() {
21         super.viewDidLoad()
22         tableView.delegate = self
23         tableView.dataSource = self
24         tableView.isHidden = false
25     }
26
27     override func viewWillAppear(_ animated: Bool) {
28         super.viewWillAppear(animated)
29         self.fetch { (complete) in
30             if complete {
31                 if goals.count >= 1 {
32                     tableView.isHidden = false
33                 } else {
34                     tableView.isHidden = true
35                 }
36             }
37         }
38     }
}
```

identity and Type  
Name: GoalsVC.swift  
Type: Default - Swift Source  
Location: Relative to Group  
Full Path: /Users/calebstultz/Documents/Devs/ios/Courses/iOS 11/Core Data/App/goalpost-dev/goalpost-app/goalpost-app/Controller/GoalsVC.swift

On Demand Resource Tags  
Only resources are taggable

Target Membership  
 goalpost-app

Text Settings  
Text Encoding: No Explicit Encoding  
Line Endings: No Explicit Line Endings  
Indent Using: Spaces  
Widths: Tab 4 Indent 4  
 Wrap lines

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

View - Represents a rectangular region in which it draws and receives events.

The screenshot shows an Xcode interface with a video overlay of Caleb Stultz. The code being discussed is from the `GoalsVC.swift` file, specifically the `func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell` implementation.

```
        return }
    presentDetail(createGoalVC)

}

extension GoalsVC: UITableViewDelegate, UITableViewDataSource {
    func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

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

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        guard let cell = tableView.dequeueReusableCell(withIdentifier: "goalCell") as? GoalCell else { return UITableViewCell() }
        let goal = goals[indexPath.row]
        cell.configureCell(description: goal.goalDescription, type: goal.goalType, goalProgressAmount: goal.goalProgress)
        return cell
    }
}

extension GoalsVC {
    func fetch(completion: (_ complete: Bool) -> ()) {
        guard let managedContext = appDelegate?.persistentContainer.viewContext else { return }

        let fetchRequest = NSFetchRequest(entityName: "Goal")
    }
}
```

The code highlights the `cellForRowAt` implementation, which dequeues a reusable cell with the identifier "goalCell", checks if it's a `GoalCell`, and then configures it with the goal's description, type, and progress amount.

On the right side of the Xcode interface, the `Identity and Type` inspector shows the file is named `GoalsVC.swift` and is a `Default - Swift Source`. It also lists the full path and target membership.

The bottom right corner of the Xcode window contains three callout boxes with descriptions:

- View Controller** - A controller that manages a view.
- Storyboard Reference** - Provides a placeholder for a view controller in an external storyboard.
- View** - Represents a rectangular region in which it draws and receives events.

The bottom status bar shows the date and time as 2017-07-31 16:38:50, and the output pane shows a log message indicating successful data saving.