

Hello Watch! Build your First Apple Watch App

Kristina Fox
Senior iOS Engineer, Intuit
@krstnfx

**Download project files:
bit.ly/ghcwatch**



Hi, I'm Kristina!



Workshop Agenda

1. Intro to Apple Watch Development
2. Counter App with iPhone Companion App
3. Apple Ecosystem
4. Resources and Q&A

Apple Watch

Apple Watch



People are more interested in the Apple Watch now than they were in the first iPhone after it launched



Lisa Eadicicco [✉](#) [🐦](#)

⌚ Jun. 24, 2015, 9:48 AM [🔥 3,715](#)

[FACEBOOK](#)

[LINKEDIN](#)

[TWITTER](#)

[EMAIL](#)

[PRINT](#)

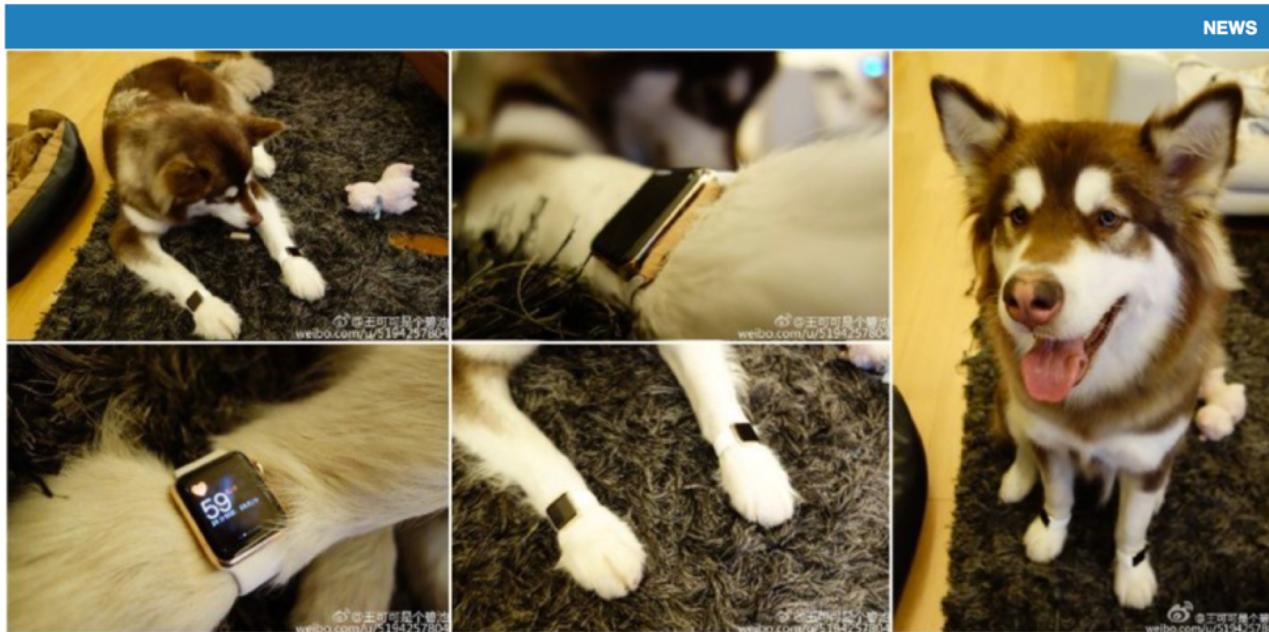
The Apple Watch officially launched in April, and it looks like people are still eager to get their hands on one, according to the latest research note from Morgan Stanley.

The firm tracked demand for the Apple Watch through its AlphaWise survey, noting that interest in the Apple Watch five to six weeks following its launch is about 20% higher than it was for



World's most spoiled dog scores two Apple Watch Editions

BY [EVAN KILLHAM](#) • 4:15 PM, MAY 26, 2015



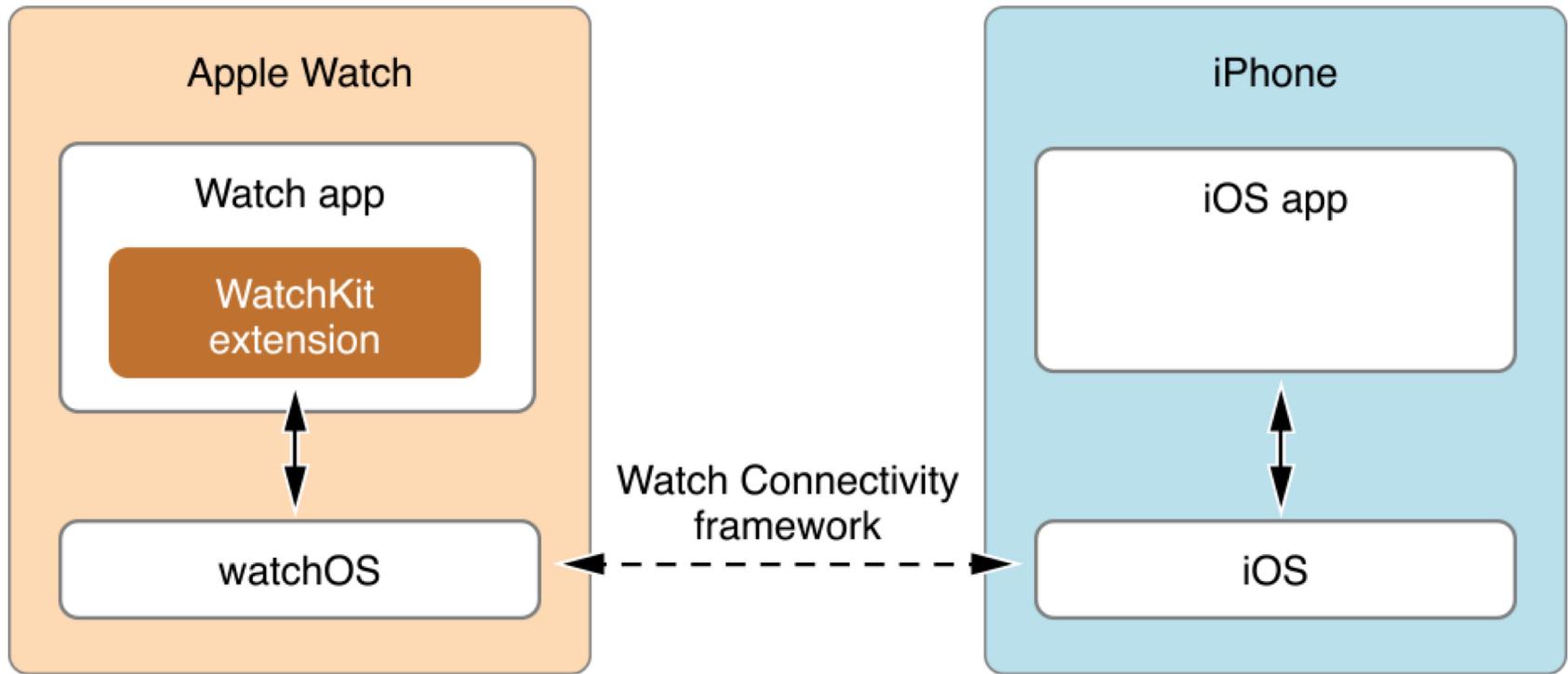
Above: The oblivious new owner of two new Apple Watch Editions.

Photo: Wang Sicong

The son of China's richest man has bought not one, but two gold Apple Watch Editions. For his dog.

Apple Watch Communication Architecture

Apple Watch Communication Architecture



Hello Watch! App

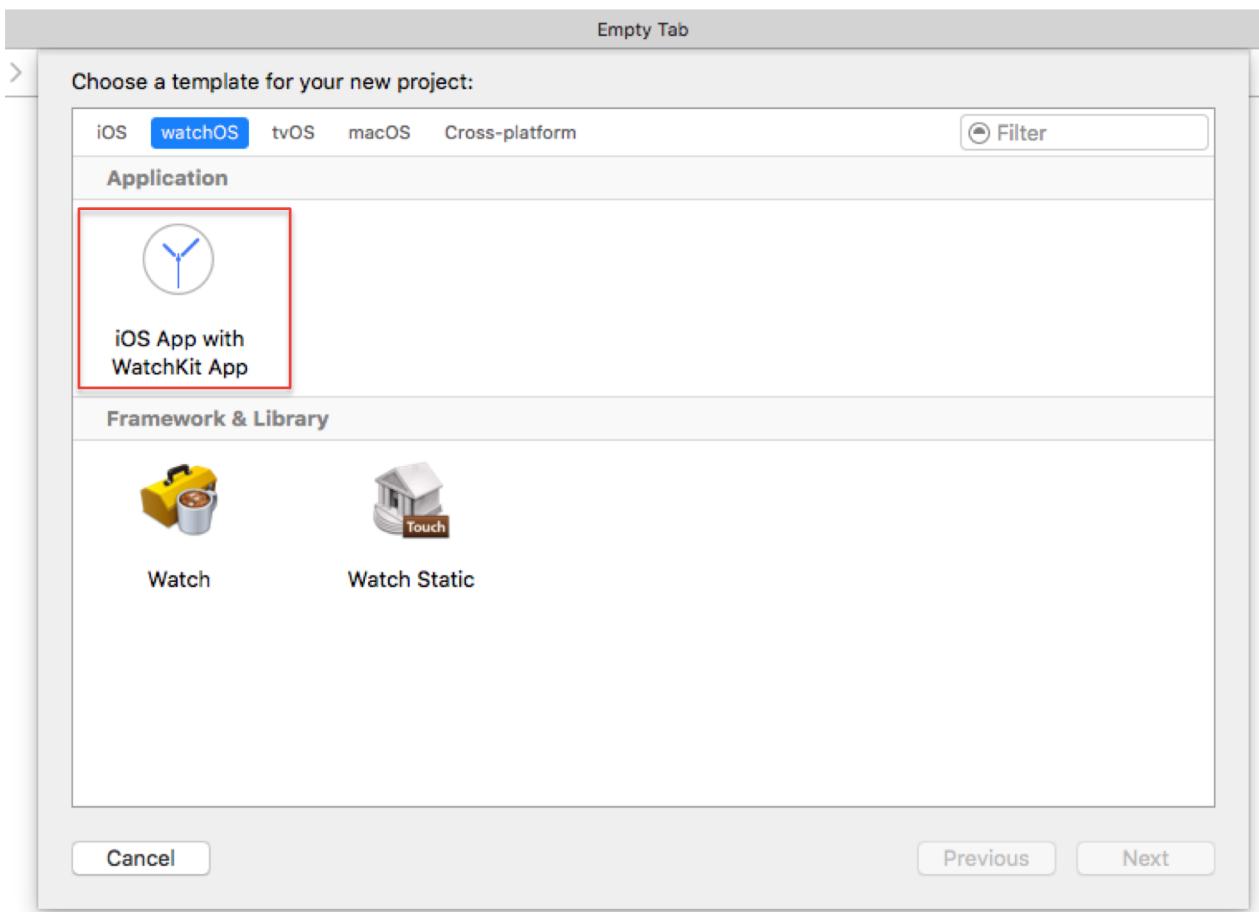
Hello Watch Application

The image shows the Xcode 8.0 welcome screen. On the left, there's a large icon of a hammer and a ruler on a blueprint. Below it, the text "Welcome to Xcode" and "Version 8.0 (8A218a)" are displayed. On the right, a sidebar lists three recent projects: "QuickBooks Self-Employed" (with path "~/Developer/rocoto"), "try Swift" (with path "~/Dropbox (Personal)/Projects"), and "Practice" (with path "~/Dropbox (Personal)/Projects"). At the bottom right of the main area, there's a link "Open another project...".

-  **Get started with a playground**
Explore new ideas quickly and easily.
-  **Create a new Xcode project**
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV. (This item is highlighted)
-  **Check out an existing project**
Start working on something from an SCM repository.

[Open another project...](#)

Choose App Type



Name Your Project

Empty Tab

Choose options for your new project:

Product Name: HelloWatch

Team: None

Organization Name:

Organization Identifier: kristina

Bundle Identifier: kristina.HelloWatch

Language: Swift

Devices: Universal

Include Notification Scene

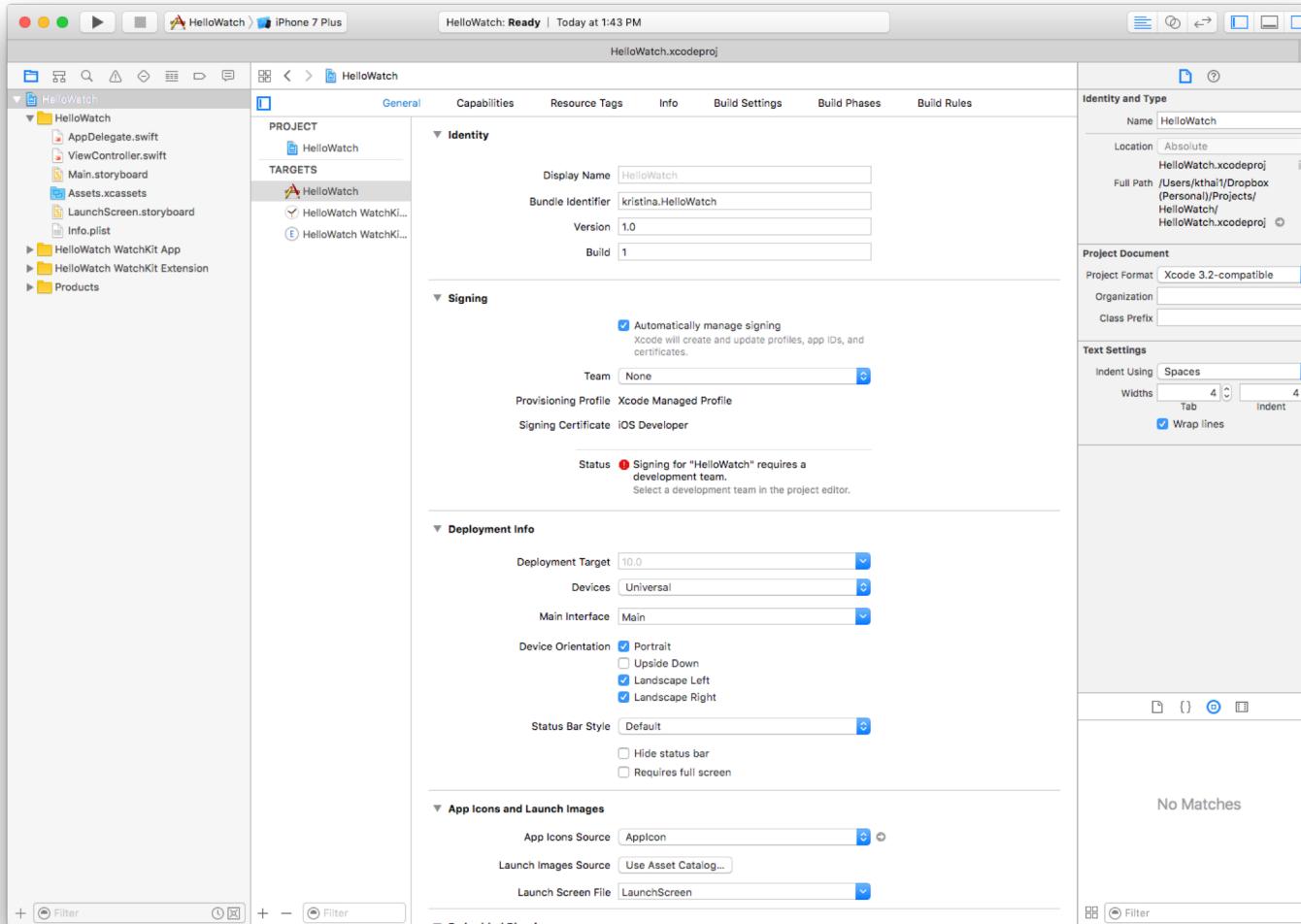
Include Complication

Include Unit Tests

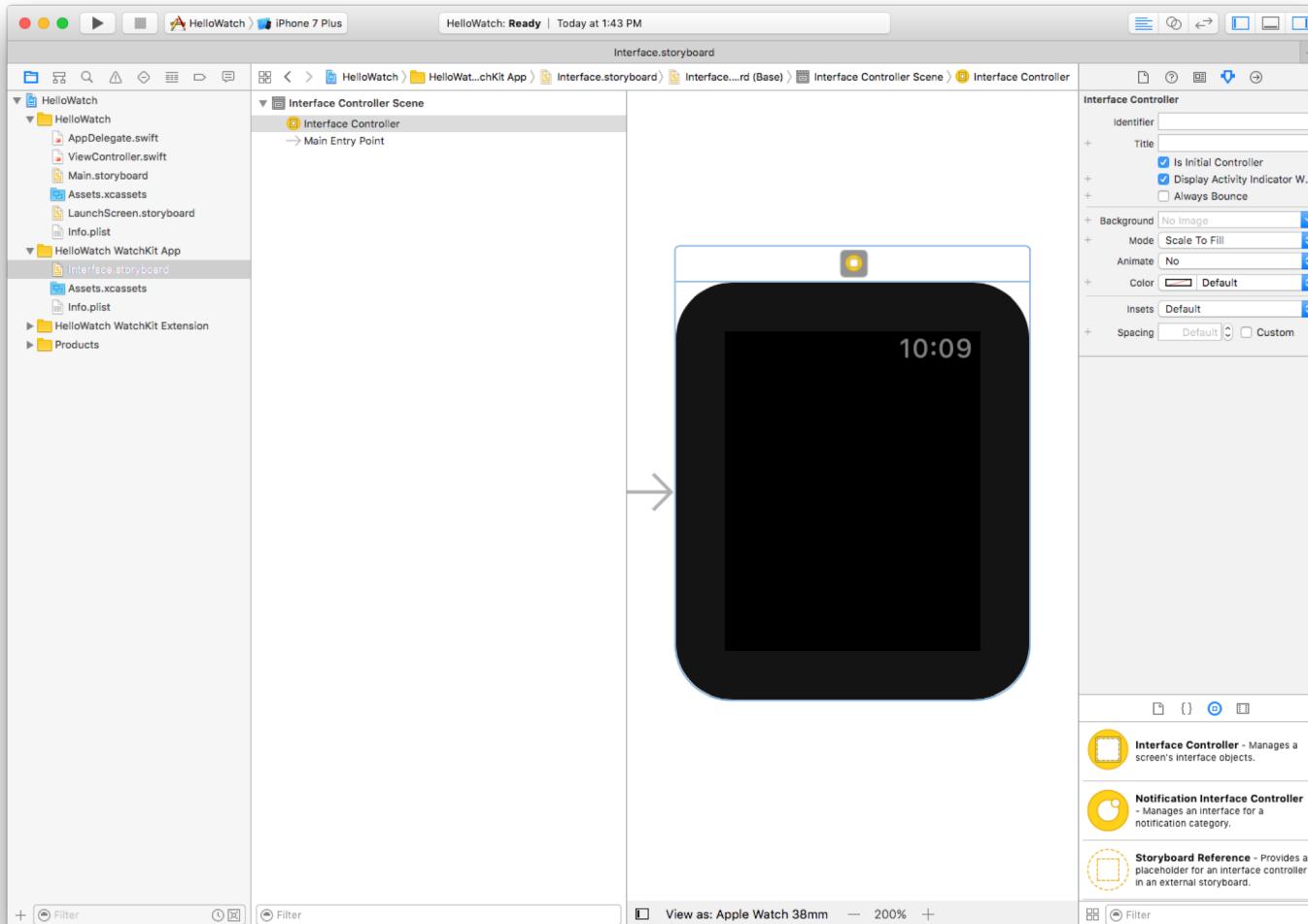
Include UI Tests

[Cancel](#) [Previous](#) [Next](#)

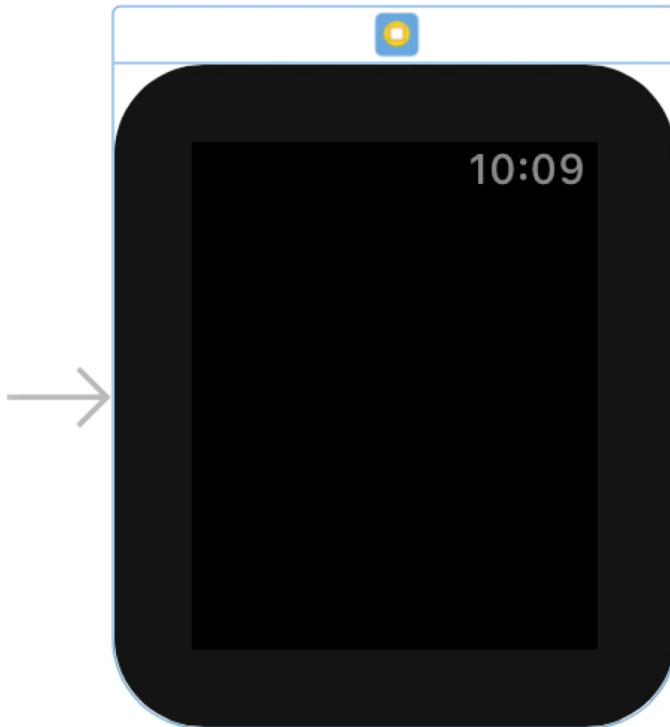
You're In!



Build the Interface



Drag the Label



Background No Image

Mode Scale To Fill

Animate No

Color Default

Insets Default

Spacing Default

Slider - A control for selecting a floating-point value from a range of continuous or discrete values.

Picker - A control for selecting an item from a list.

Label Label - Displays a static text string.

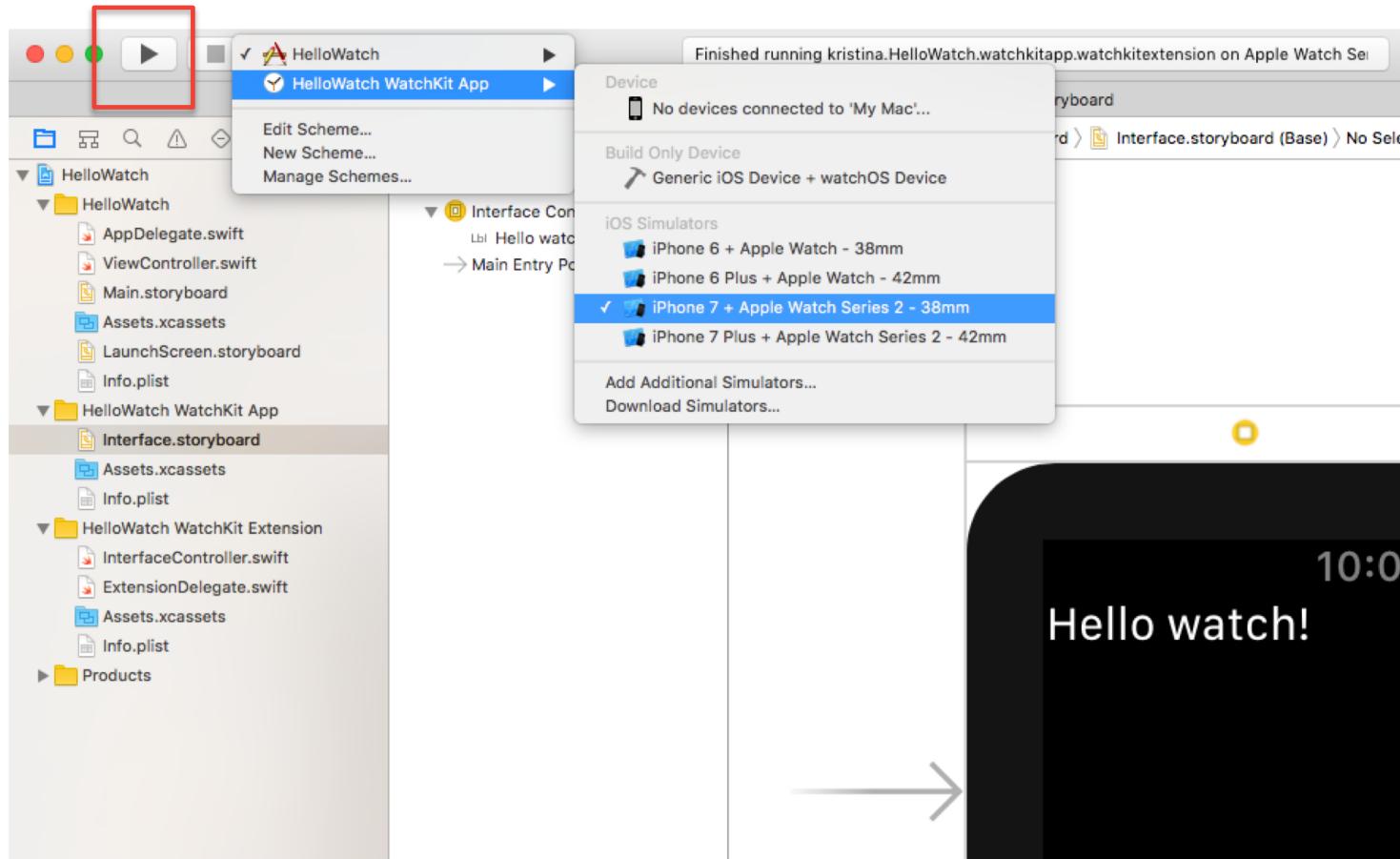
9/9/14 Date - Displays the current date and time.

59:59 Timer - Displays a string that counts up or down to a specified time.

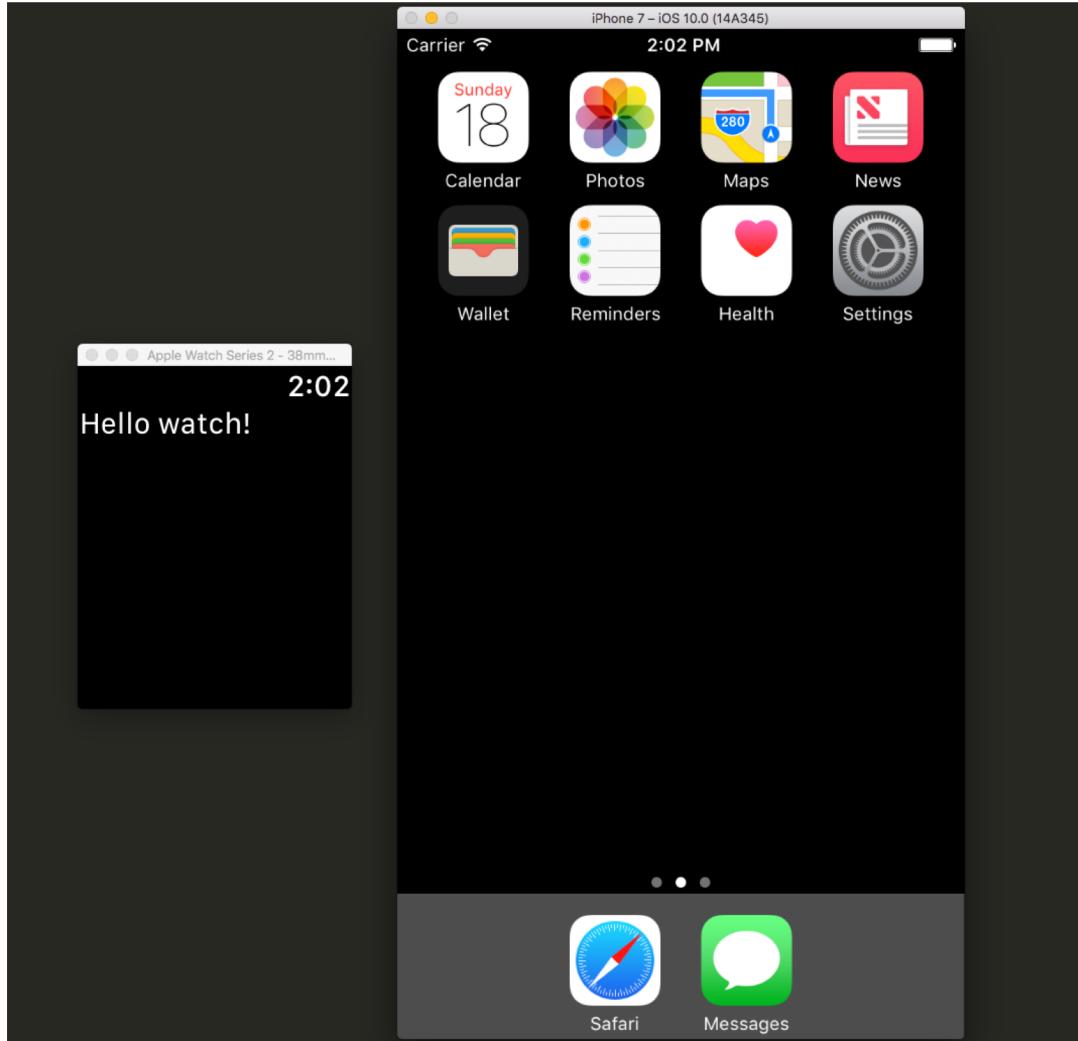
Activity Ring Activity Ring - Displays an Activity Ring

A red arrow points from the text 'Drag the Label' towards the 'Label' section in the sidebar.

Pick a Scheme...and Run!

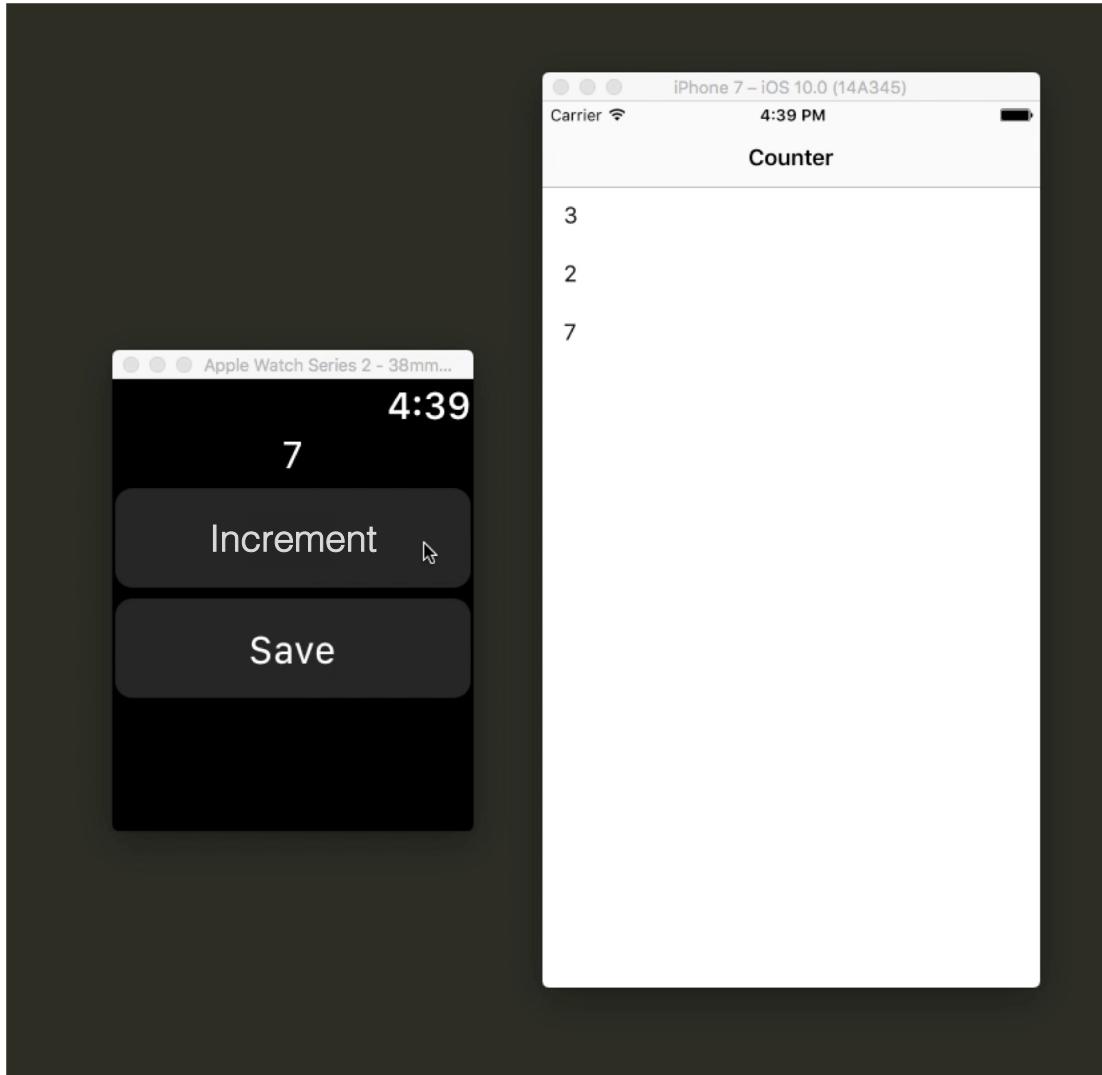


Launch the App!



Watch Counter App (with iPhone Companion App)

Final Product

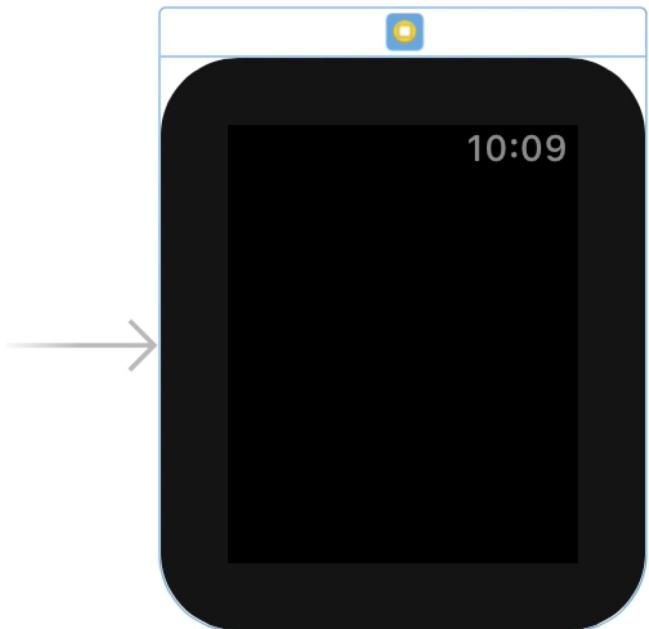


Counter Application

Open the Xcode project (Counter.xcodeproj)

bit.ly/ghcwatch

Setup the Counter Label



Interface Controller

Identifier:

+ Title: Is Initial Controller

+ Display Activity Indicator When Applescript Runs:

+ Always Bounce:

+ Background: No Image

+ Mode: Scale To Fill

+ Animate: No

+ Color: Default

+ Insets: Default

+ Spacing: Default Custom

Switch - A control for indicating a binary value.

Slider - A control for selecting a floating-point value from a range of continuous or discrete values.

Picker - A control for selecting an item from a list.

Label **Label** - Displays static text string.

Add the buttons

No Selection

Separator - A line for separating content in your interface.

Button - Button - A tappable area with a title and image.

Payment Button - Standard button for initiating Apple Pay transactions.

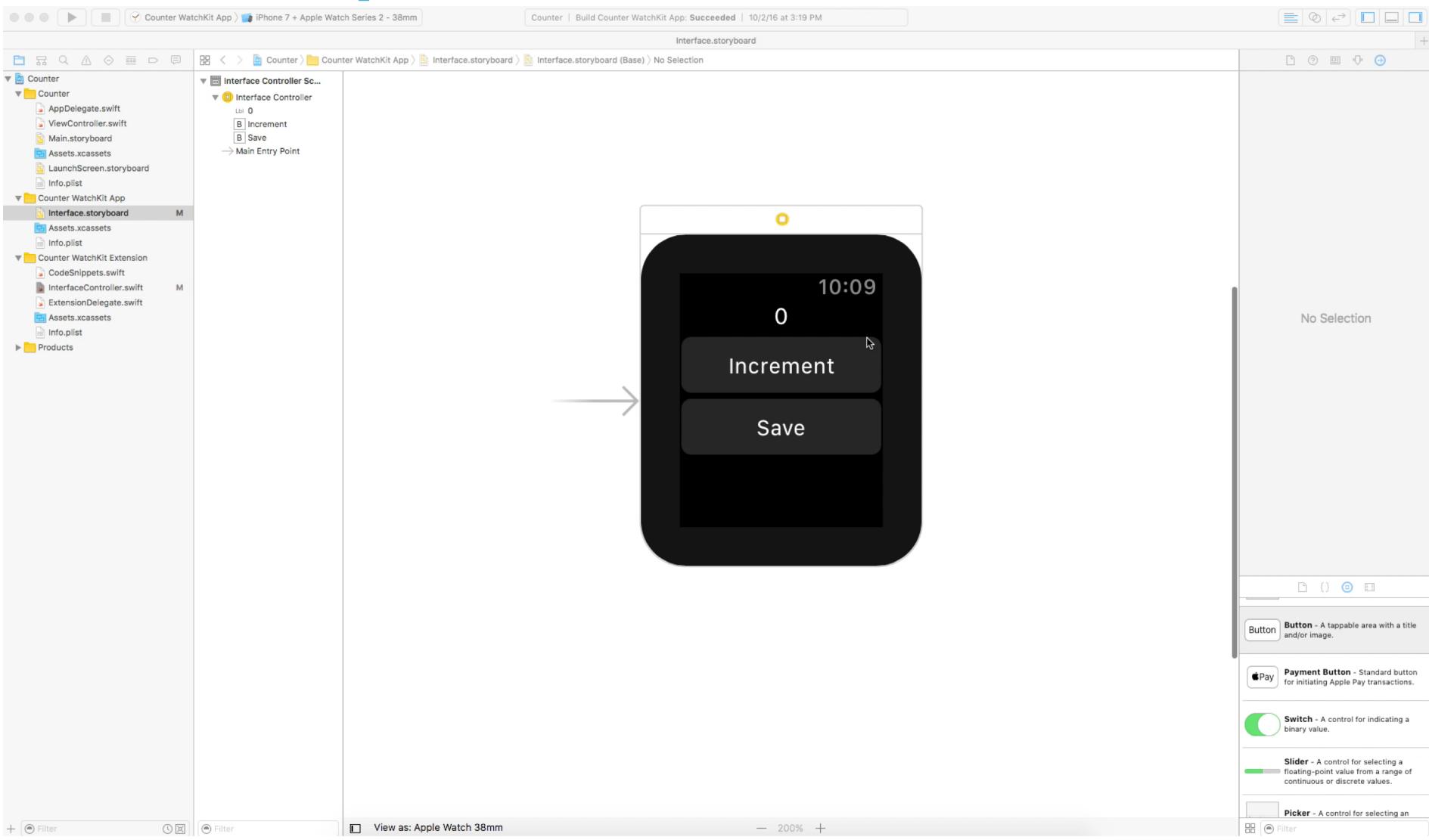
Switch - A control for indicating a binary value.

Slider - A control for selecting a floating-point value from a range of continuous or discrete values.

Picker - A control for selecting an item from a list.

Label - Label - Displays a static text string.

Go into split view



Connect the counter label outlet

The screenshot shows two Xcode windows side-by-side. On the left is the 'Interface.storyboard' window, which displays a WatchKit interface with a digital clock showing 10:09 and a counter labeled '0'. Below the counter are two buttons: 'Increment' and 'Save'. A large grey arrow points from the storyboard towards the code editor on the right. On the right is the 'InterfaceController.swift' file, showing the following Swift code:

```
1 //  
2 // InterfaceController.swift  
3 // Counter WatchKit Extension  
4 //  
5 // Created by Thai, Kristina on 9/18/16.  
6 //  
7 //  
8  
9 import WatchKit  
10 import Foundation  
11 /*  
12 * 2 - Import Watch Connectivity framework  
13 */  
14  
15 /*  
16 * 3 - Add WCSessionDelegate to class definition  
17 */  
18 class InterfaceController: WKInterfaceController {  
19 /*  
20 * 1 - Create a variable for your counter  
21 */  
22  
23 /*  
24 * 4 - Set up Watch Connectivity  
25 */  
26  
27 /*  
28 * 5 - Setup your incrementCounter() function  
29 */  
30  
31 /*  
32 * 6 - Setup your saveCounter() function  
33 */  
34  
35 }
```

Connect the button actions

The screenshot shows the Xcode interface with two main panes. On the left, the 'Interface.storyboard' file is displayed, showing a dark-themed watch face with a digital clock at the top showing '10:09' and a numerical counter below it set to '0'. Two buttons are present: a light gray 'Increment' button and a darker gray 'Save' button. A large white arrow points from the storyboard towards the code editor on the right. On the right, the 'InterfaceController.swift' file is shown, containing the following Swift code:

```
1 // InterfaceController.swift
2 // Counter WatchKit Extension
3 // Created by Thai, Kristina on 9/18/16.
4 //
5 import WatchKit
6 import Foundation
7 /*
8 * 2 - Import Watch Connectivity framework
9 */
10 /*
11 * 3 - Add WCSessionDelegate to class definition
12 */
13 class InterfaceController: WKInterfaceController {
14     @IBOutlet var counterLabel: WKInterfaceLabel!
15     /*
16     * 1 - Create a variable for your counter
17     */
18     /*
19     * 4 - Set up Watch Connectivity
20     */
21     /*
22     * 5 - Setup your incrementCounter() function
23     */
24     /*
25     * 6 - Setup your saveCounter() function
26     */
27 }
28
29
30
31
32
33
34
35
36
37 }
```

Time to code!

On the left sidebar, go to the Counter WatchKit Extension folder and click on "CodeSnippets.swift"

Hold down the Option key and click on "InterfaceController.swift"

```
// CodeSnippets.swift
// Counter WatchKit Extension
//
// Created by Thai, Kristina on 9/18/16.
//
/*
 * 1 - Create a variable for your counter
 */
var counter = 0

/*
 * 2 - Import Watch Connectivity framework
 */
import WatchConnectivity

/*
 * 3 - Add WCSessionDelegate to class definition
 */
class InterfaceController: WKInterfaceController, WCSessionDelegate {

/*
 * 4 - Set up Watch Connectivity
 */

// InterfaceController.swift
// Counter WatchKit Extension
//
// Created by Thai, Kristina on 9/18/16.
//
import WatchKit
import Foundation
/*
 * 2 - Import Watch Connectivity framework
 */
/*
 * 3 - Add WCSessionDelegate to class definition
 */
class InterfaceController: WKInterfaceController {
    @IBOutlet var counterLabel: WKInterfaceLabel!
    /*
     * 1 - Create a variable for your counter
     */

    /*
     * 4 - Set up Watch Connectivity
     */

    /*
     * 5 - Setup your incrementCounter() function
     */
}
```

1 - Create a variable for your counter

```
15  /*
16   * 3 - Add WCSessionDelegate to class definition
17   */
18  class InterfaceController: WKInterfaceController {
19      @IBOutlet var counterLabel: WKInterfaceLabel!
20      /*
21       * 1 - Create a variable for your counter|
22       */
23      var counter = 0
24
25      /*
26       * 4 - Set up Watch Connectivity
27       */
28
```

2 - Import Watch Connectivity framework

```
9 import WatchKit
10 import Foundation
11 /*
12 * 2 - Import Watch Connectivity framework
13 */
14 import WatchConnectivity
15
16 /*
17 * 3 - Add WCSessionDelegate to class definition
18 */
19 class InterfaceController: WKInterfaceController {
```

3 - Add WCSessionDelegate to class definition

```
16  /*
17   * 3 - Add WCSessionDelegate to class definition
18   */
19  class InterfaceController: WKInterfaceController, WCSessionDelegate {
20      @IBOutlet var counterLabel: WKInterfaceLabel!
21      /*
22       * 1 - Create a variable for your counter
23       */
24      var counter = 0
25
26      /*
27       * 4 - Set up Watch Connectivity
28       */
29
```

4 - Setup Watch Connectivity

```
25
26  /*
27   * 4 - Set up Watch Connectivity
28   */
29  private let session : WCSession? = WCSession.isSupported() ?
30      WCSession.default() : nil
31
32  override init() {
33      super.init()
34
35      session?.delegate = self
36      session?.activate()
37  }
38
39  func session(_ session: WCSession, activationDidCompleteWith
40      activationState: WCSessionActivationState, error: Error?) {}
41
42  /*
43   * 5 - Setup your incrementCounter() function
44   */
45  @IBAction func incrementCounter() {
46  }
```

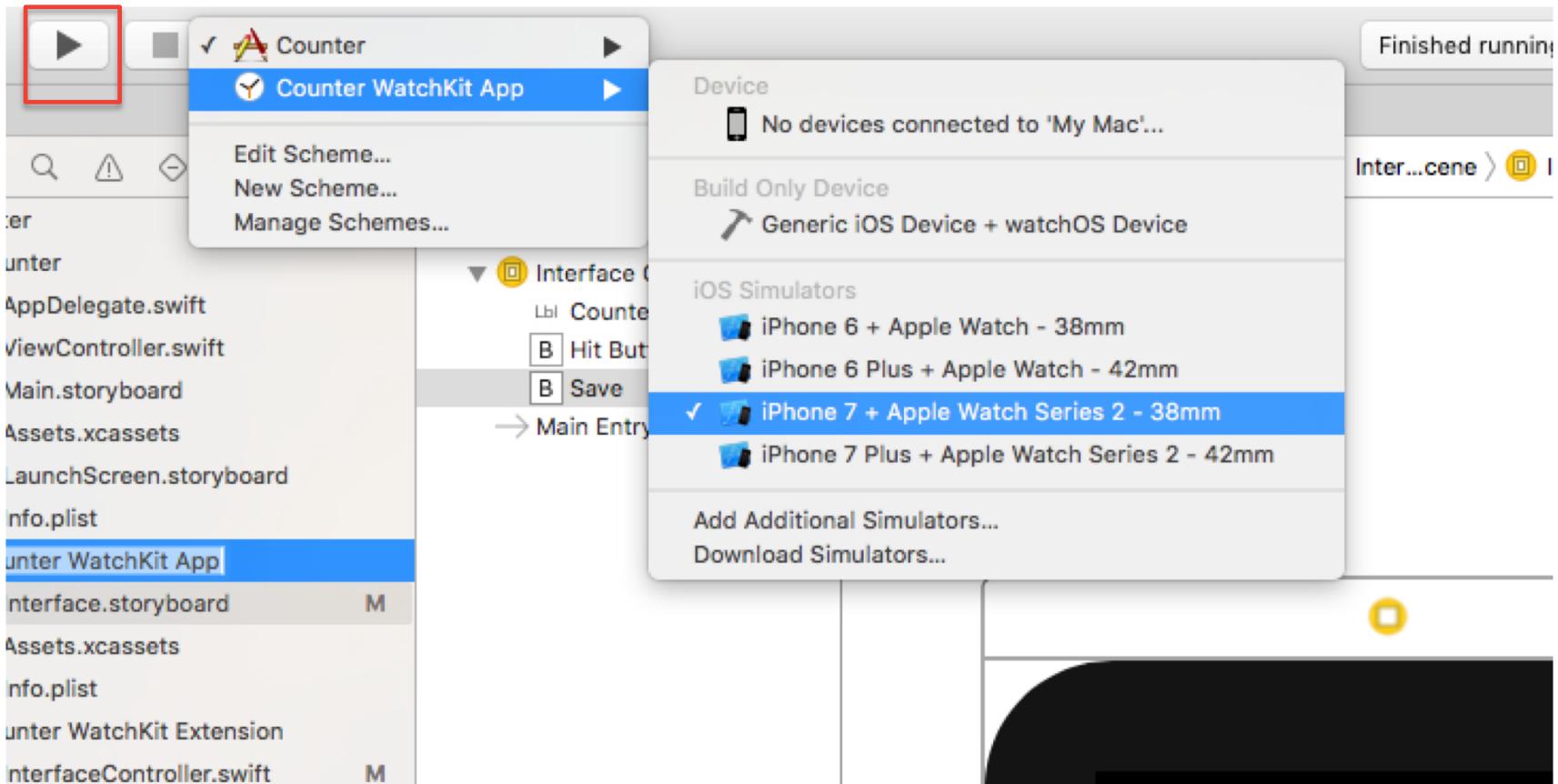
5 - Setup your incrementCounter() function

```
38 func session(_ session: WCSession, activationDidCompleteWith  
activationState: WCSessionActivationState, error: Error?) {}  
39  
40 /*  
41 * 5 - Setup your incrementCounter() function  
42 */  
④ 43 @IBAction func incrementCounter() {  
44     counter+=1;  
45     counterLabel.setText(String(counter))  
46 }  
47
```

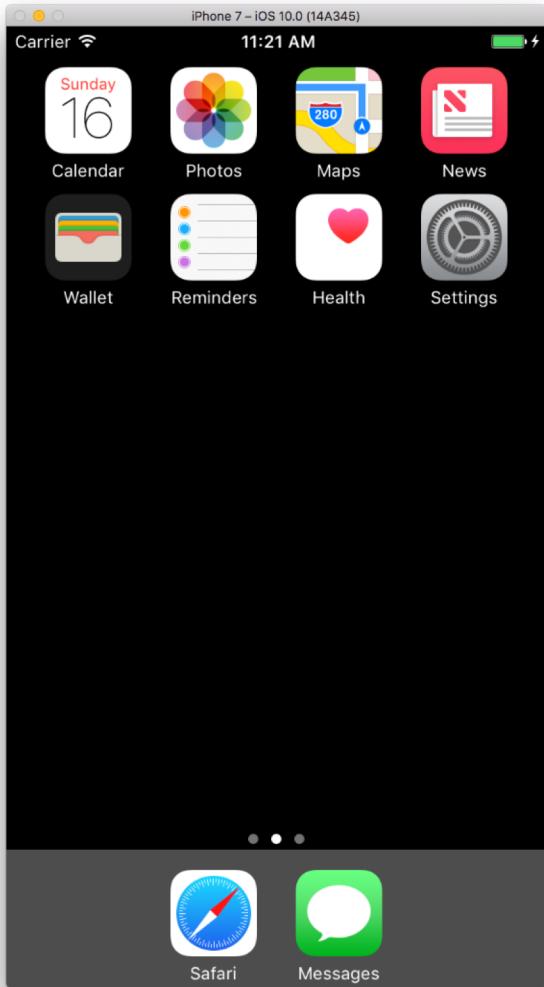
6 - Setup your saveCounter() function

```
47
48     /*
49      * 6 - Setup your saveCounter() function
50      */
51     @IBAction func saveCounter() {
52         let applicationData = ["counterValue" : counter]
53
54         if let session = session, session.isReachable {
55             session.sendMessage(applicationData, replyHandler: nil,
56                 errorHandler: { error in
57                     print(error)
58                 })
59     }
```

Choose Your Scheme...and Run!

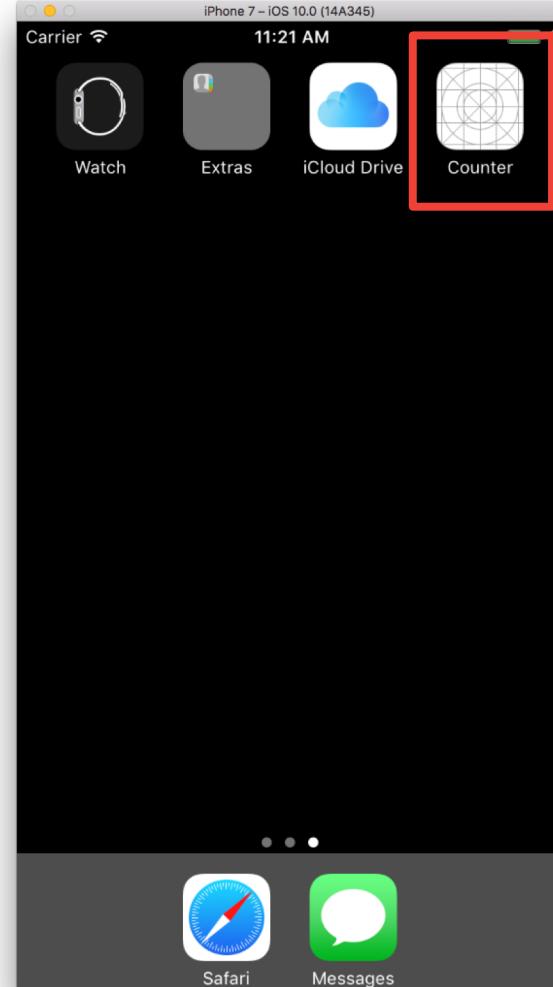


Open the Counter app on the iPhone

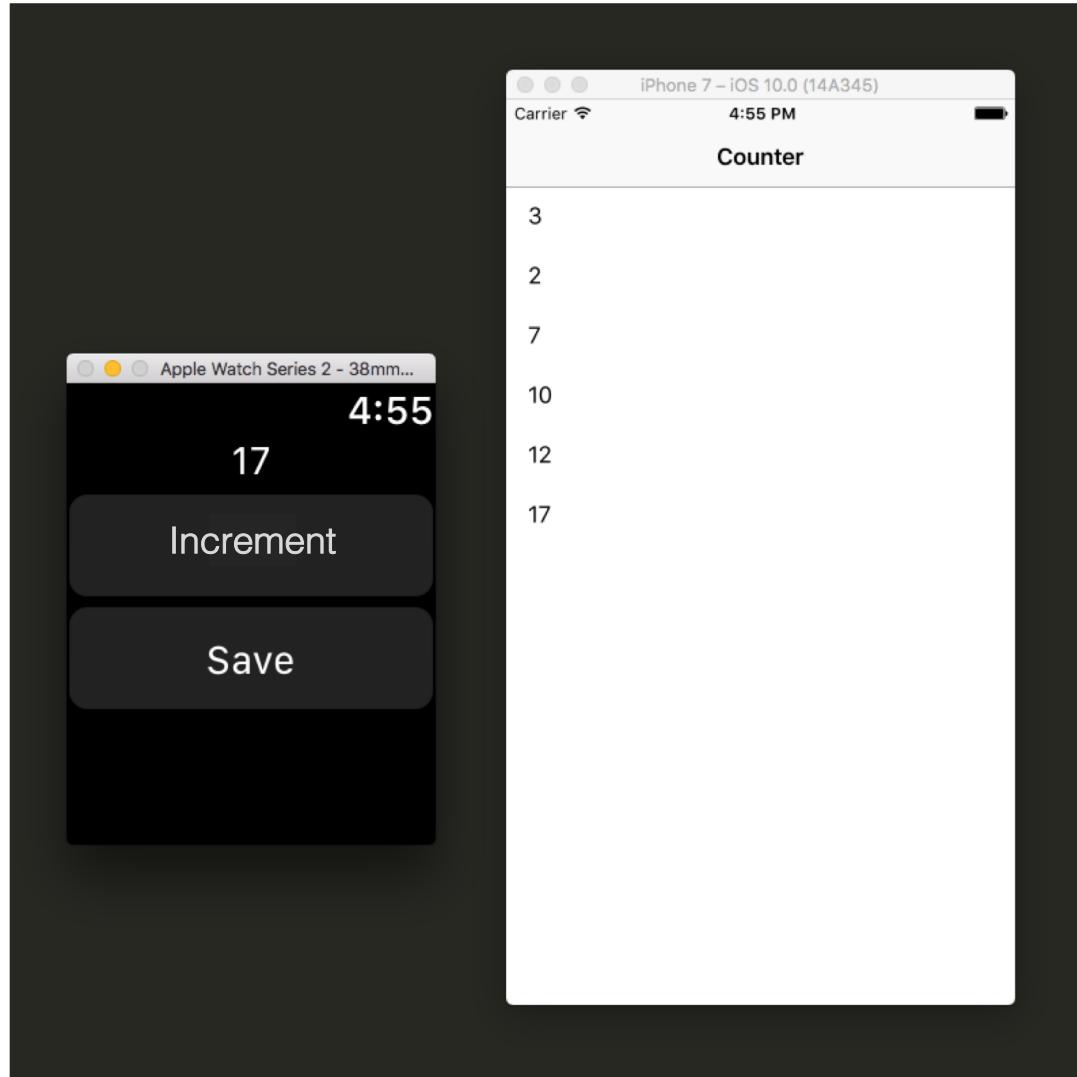


Swipe left to get
to the 2nd page

Click on the
Counter app



Counter Application



Final Project Code

This repository

Search

Pull requests Issues Gist

Unwatch 2 Unstar 2 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

12 commits 1 branch 0 releases 2 contributors MIT

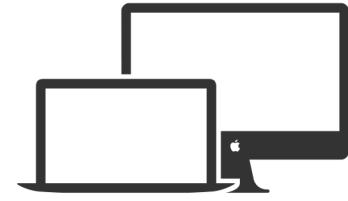
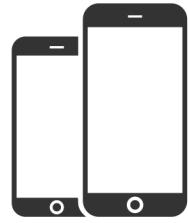
Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit Message	Time
Counter WatchKit App	Add code snippets file	a minute ago
Counter WatchKit Extension	Add code snippets file	a minute ago
Counter.xcodeproj	Add code snippets file	a minute ago
Counter	Collapse functions we won't cover in the workshop	an hour ago
LICENSE	Initial commit	14 days ago
README.md	Initial commit	14 days ago

<https://github.com/krstnfx/watchOSCounter>

Apple Ecosystem

Apple Ecosystem



Apple Ecosystem

watchOS tvOS macOS iOS

Resources and Q&A

Kristina's iOS & watchOS Blog

KRISTINA.IO

WATCHOS

July 18, 2017

WATCHOS 4: FRONTMOST APP STATE

There was certainly a trend to some of the new versions of operating systems coming out this year at WWDC. Like macOS High Sierra (which I still can't believe is the real name), watchOS 4 feels like another refinement version. It seems as if the team has been hard at work making watchOS even more ...

[CONTINUE READING](#)



ABOUT

I'm Kristina. Currently a Senior iOS engineer at Intuit. International tech speaker. Interested in watchOS, women in tech outreach, and kittens. Has nightmares about deleting derived data. [More?](#)

CATEGORIES

Career (5)

Design (1)

iOS (3)

Outreach (1)

Presentations/Speaking (2)

Resources

1. [App Programming Guide for Apple Watch](#)
2. [Apple Watch Human Interface Guidelines](#)
3. [Ray Wenderlich](#)
4. [Natasha the Robot](#)

Thanks!



Kristina Fox

@krstnfx

Kristina.io