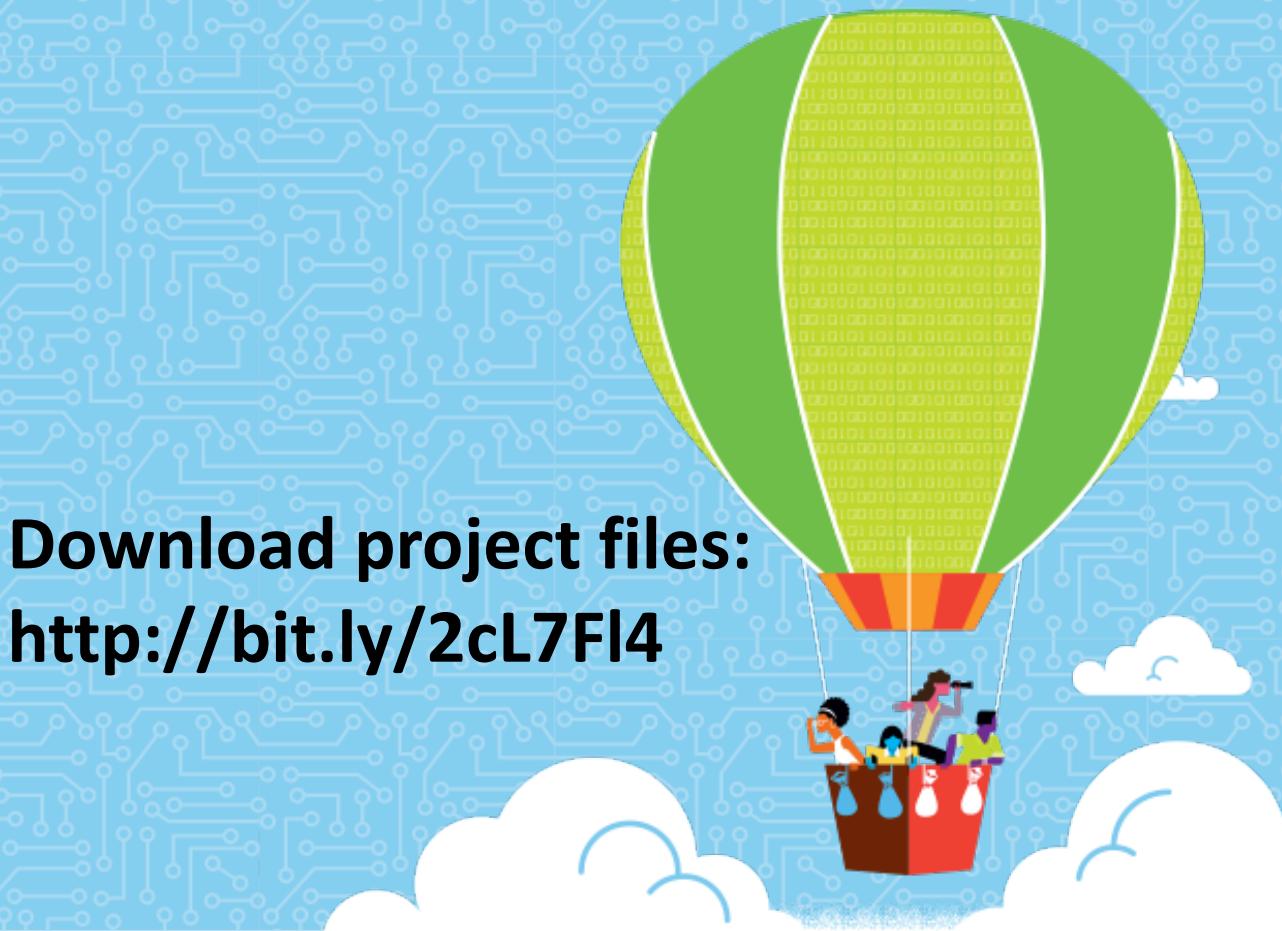


Hello Watch! Build your First Apple Watch App

Kristina Thai
iOS Software Engineer
@kristinathai

Mayuko Inoue
iOS Software Engineer
@hellomayuko



Download project files:
<http://bit.ly/2cL7FI4>

2016

ANITA BORG INSTITUTE
GRACE HOPPER CELEBRATION OF WOMEN IN COMPUTING

 #GHC16

ANITA BORG INSTITUTE 

 Association for Computing Machinery

Hi, I'm Kristina!



Hi! I'm Mayuko



Workshop Agenda

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

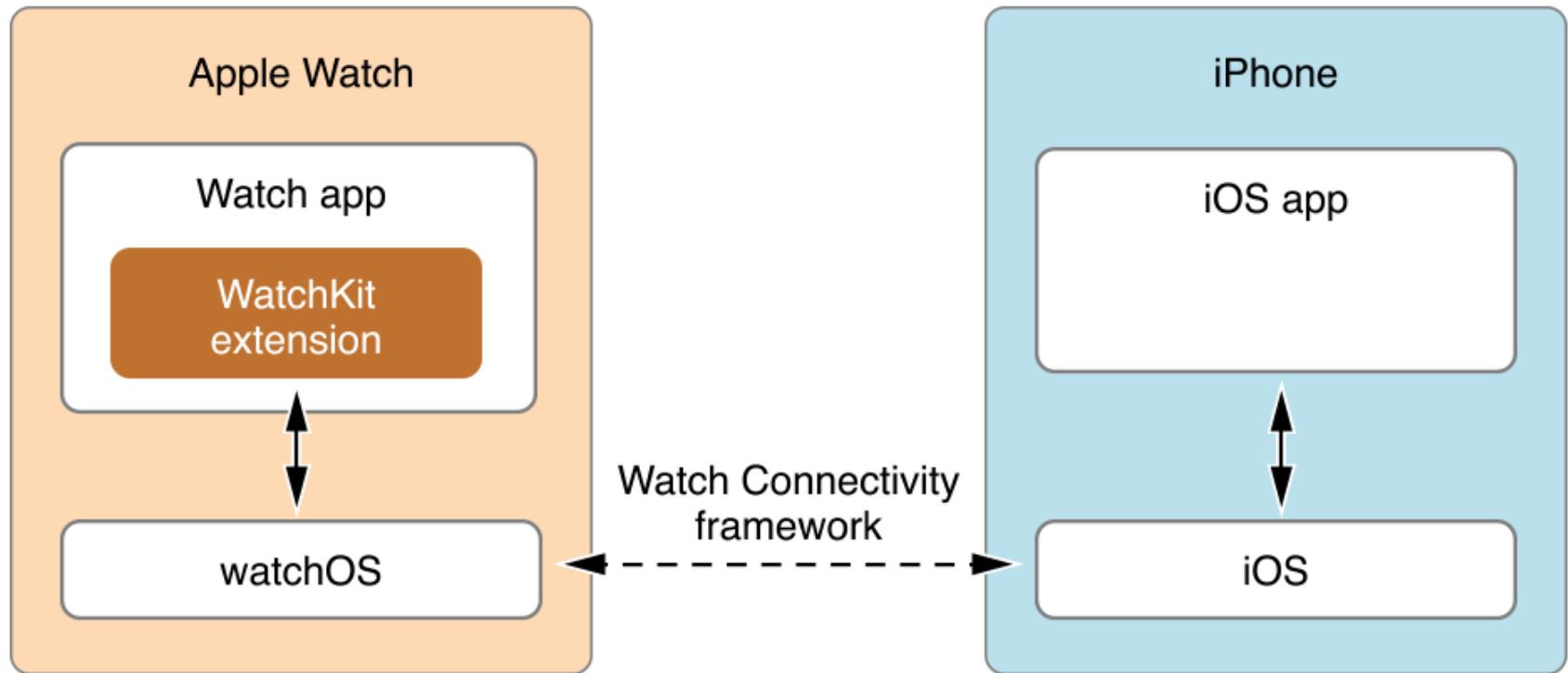
Apple Watch



Apple Watch



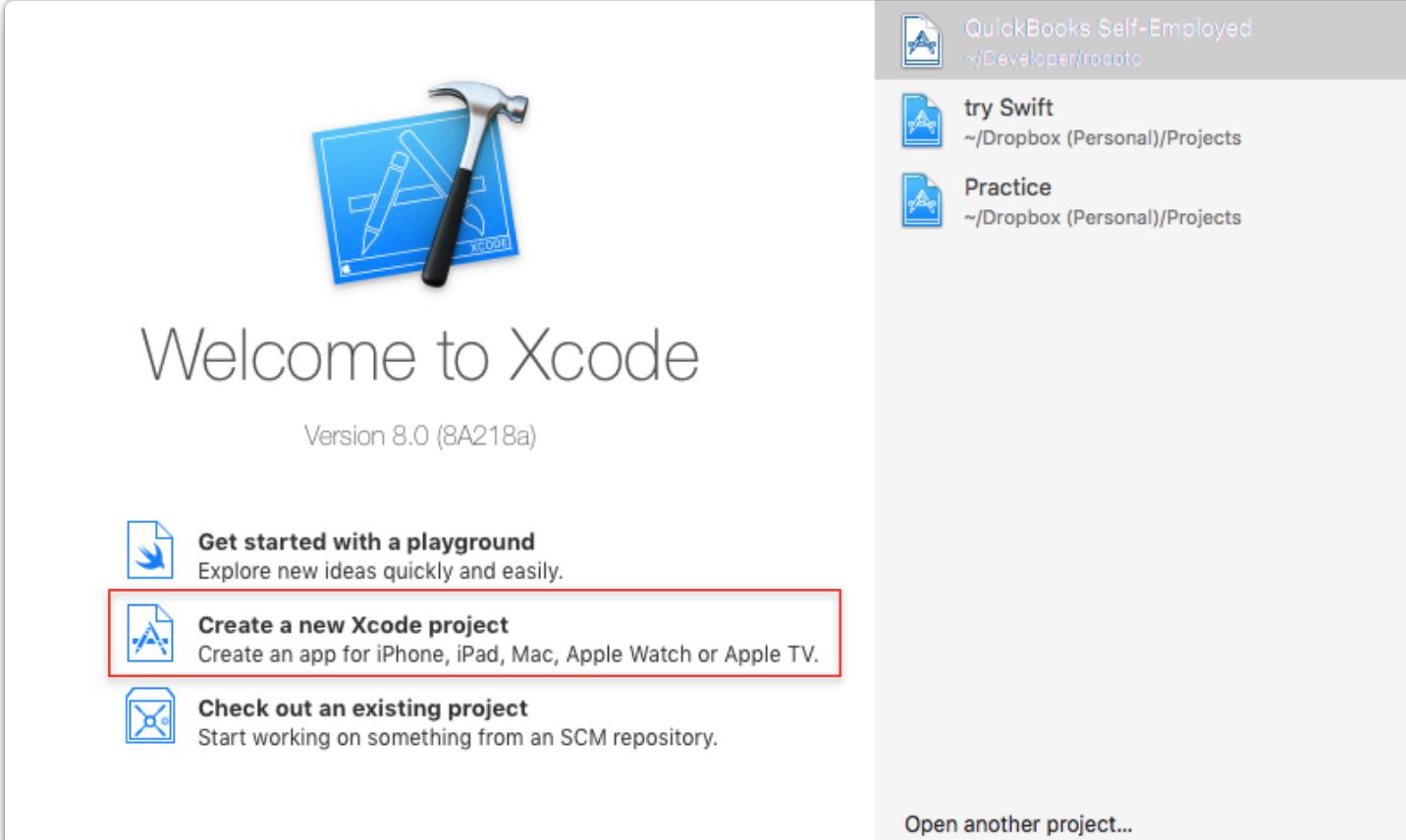
Apple Watch Architecture



Hello Watch! App



Hello Watch Application



The image shows the 'Welcome to Xcode' screen. At the top, there's a blue 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, there's a sidebar with three project suggestions: 'QuickBooks Self-Employed' (with path '~Developer/rocoto'), 'try Swift' (with path '~Dropbox (Personal)/Projects'), and 'Practice' (with path '~Dropbox (Personal)/Projects'). The bottom left contains three buttons: '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, which is highlighted with a red border), and 'Check out an existing project' (Start working on something from an SCM repository). The bottom right has a link 'Open another project...'. A decorative blue circuit board pattern is at the very top of the slide.

Welcome to Xcode

Version 8.0 (8A218a)

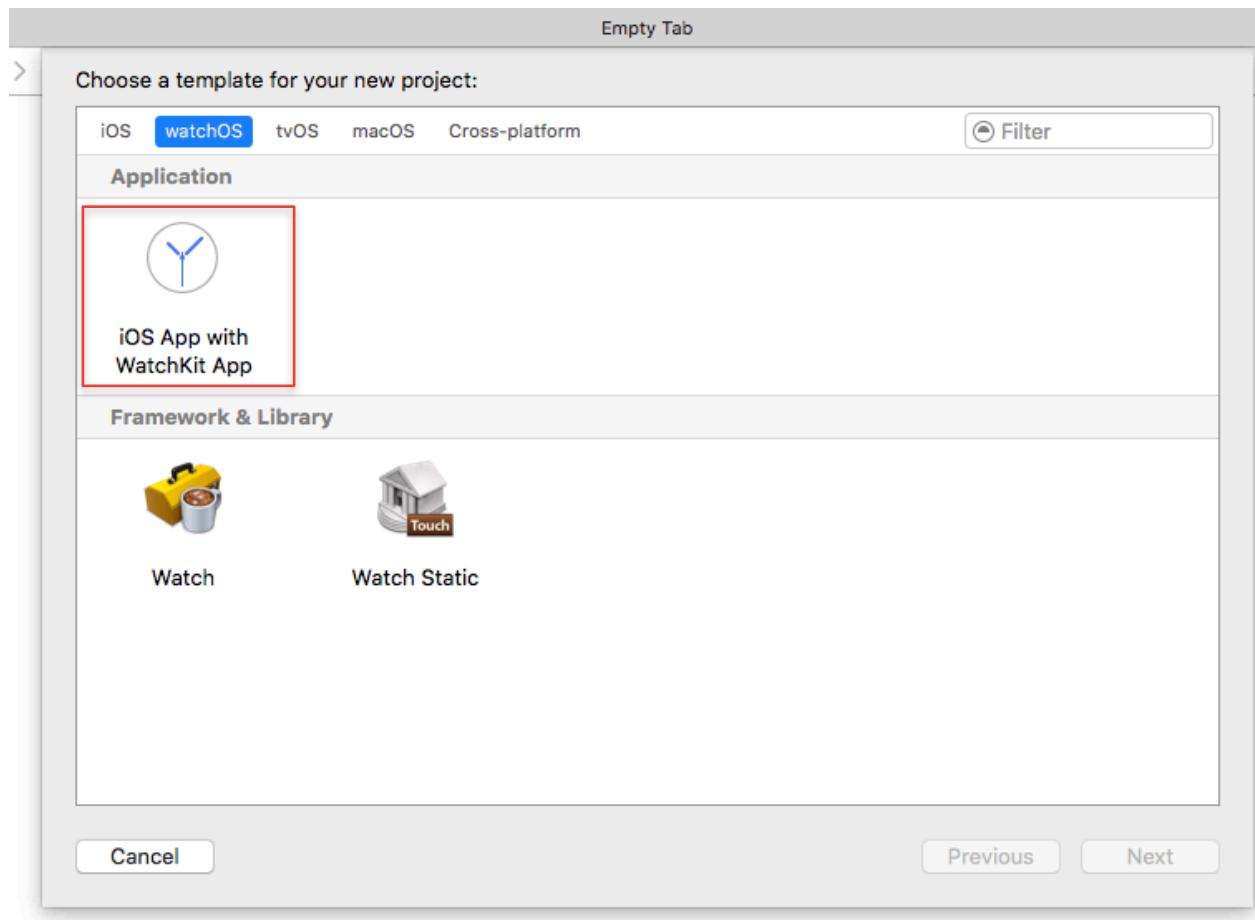
Get started with a playground
Explore new ideas quickly and easily.

Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.

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

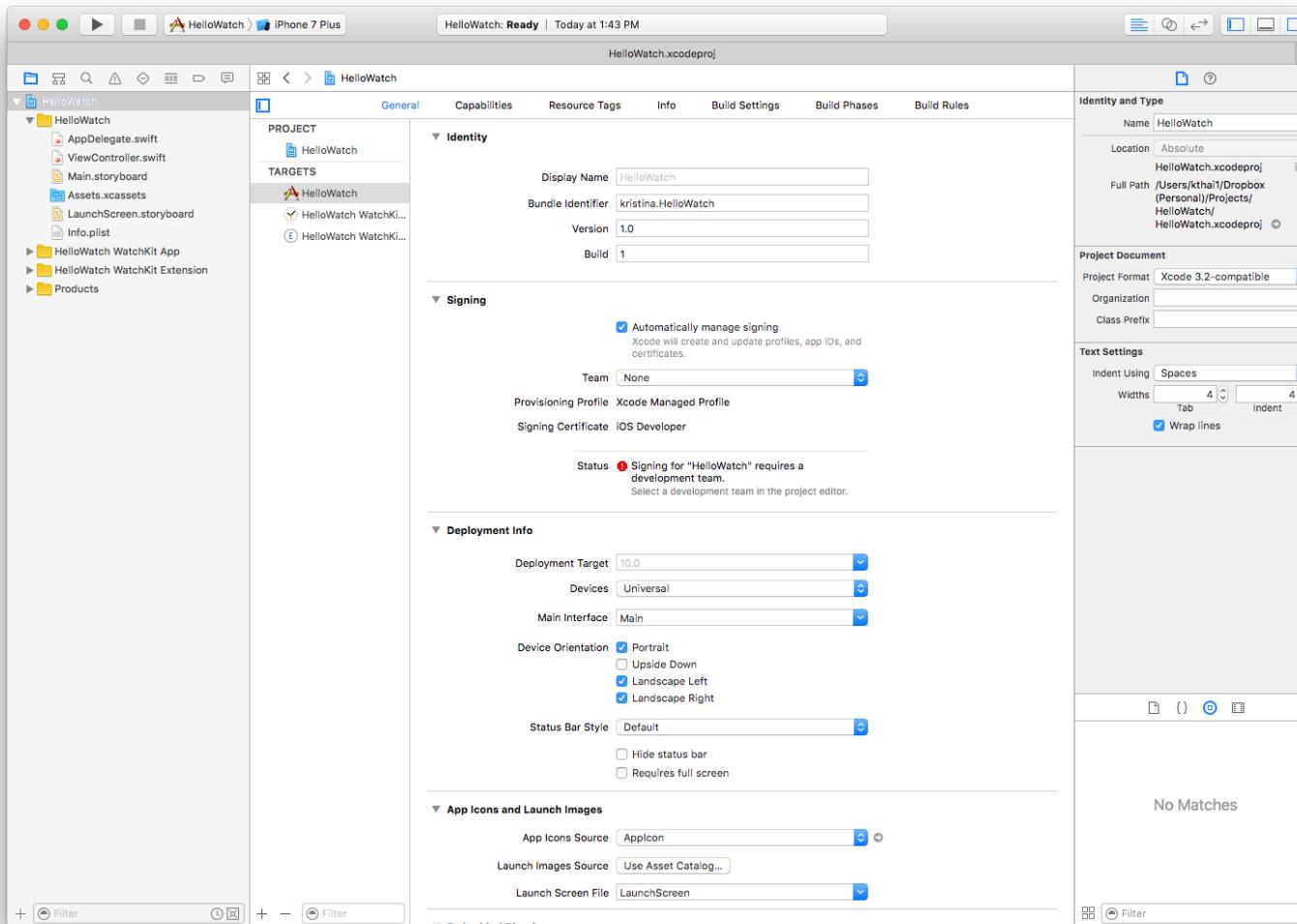
Include Complication

Include Unit Tests

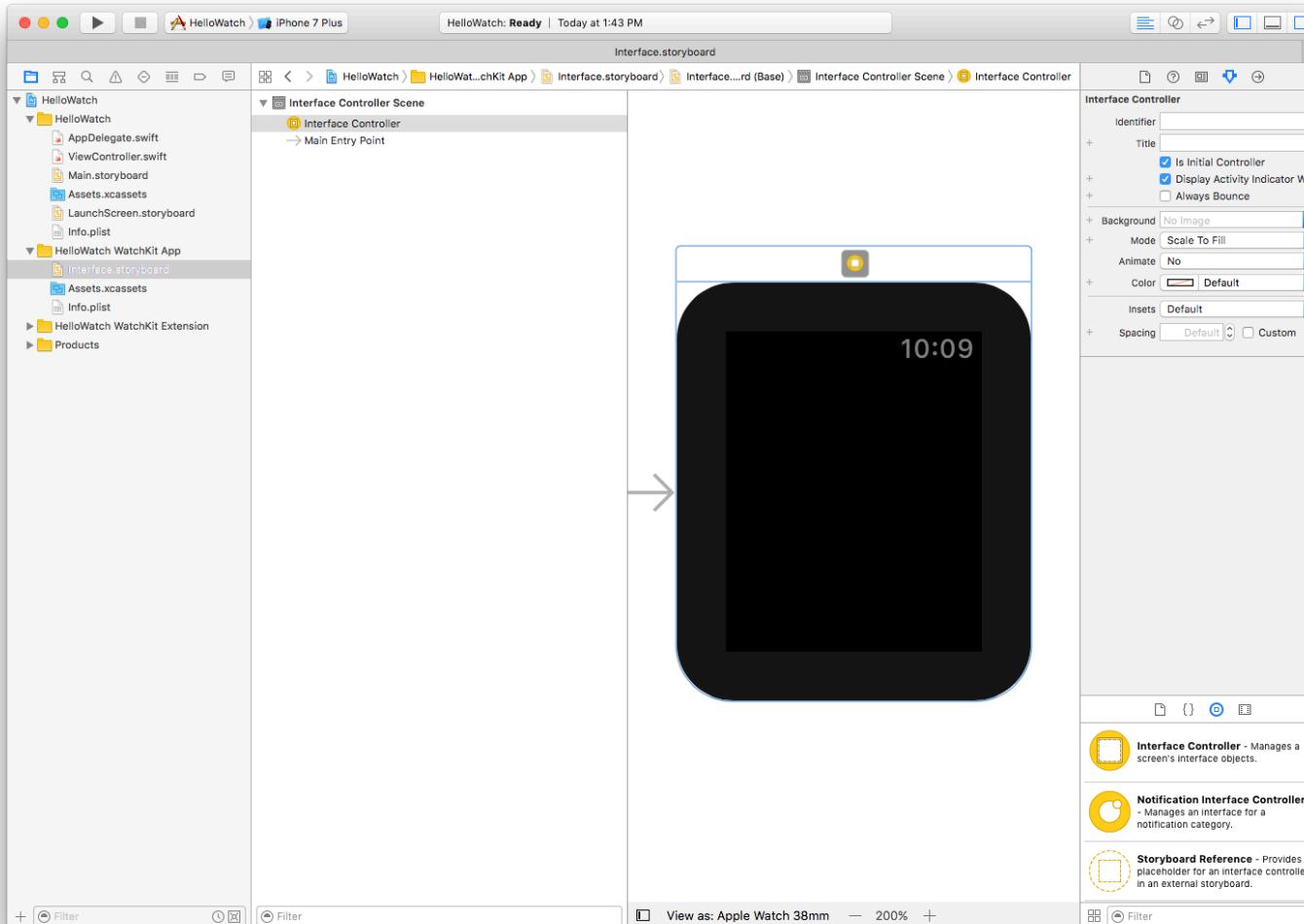
Include UI Tests

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

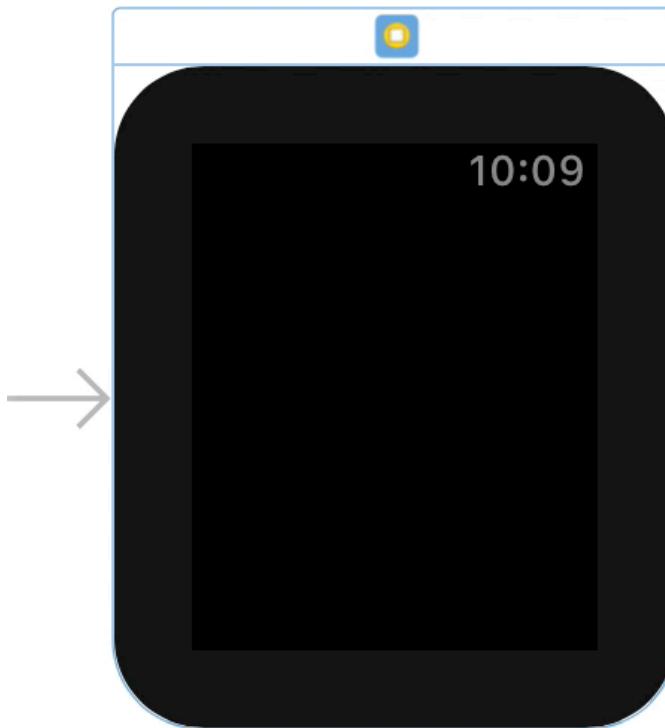
You're In!



Build the Interface



Drag the Label



The image shows the Xcode interface with the object library open on the right side. The 'Label' section is highlighted, showing a preview of a white label with black text and a description: 'Label - Label - Displays a static text string.' Below it are other controls: 'Slider', 'Picker', 'Date', 'Timer', and 'Activity Ring'. Each control has a small icon and a brief description. The 'Label' section is currently selected, indicated by a cursor icon.

+ Background No Image
+ Mode Scale To Fill
Animate No
+ Color Default
Insets Default
+ Spacing Default Custom

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.

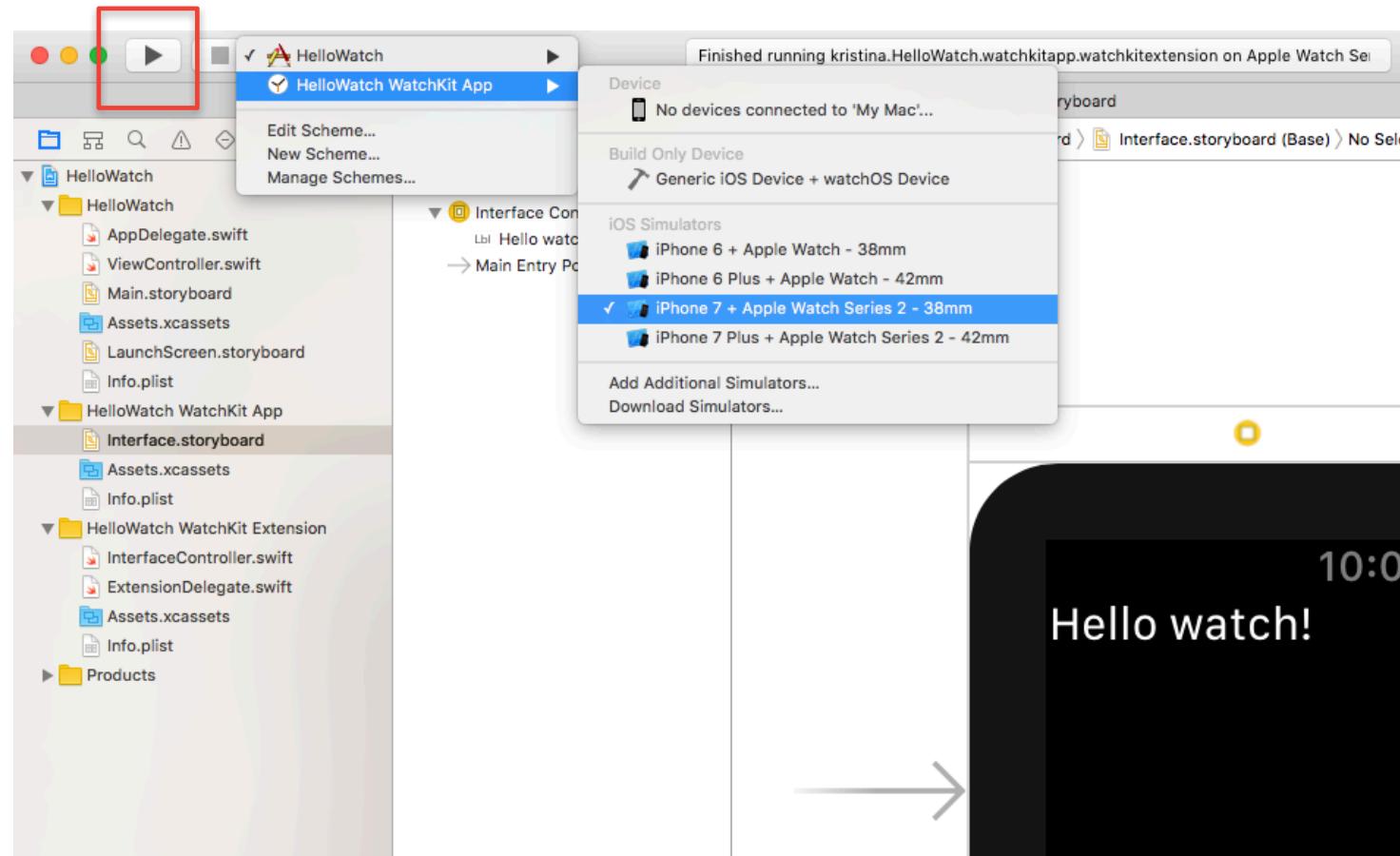
Label Label - Displays a static text string.

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

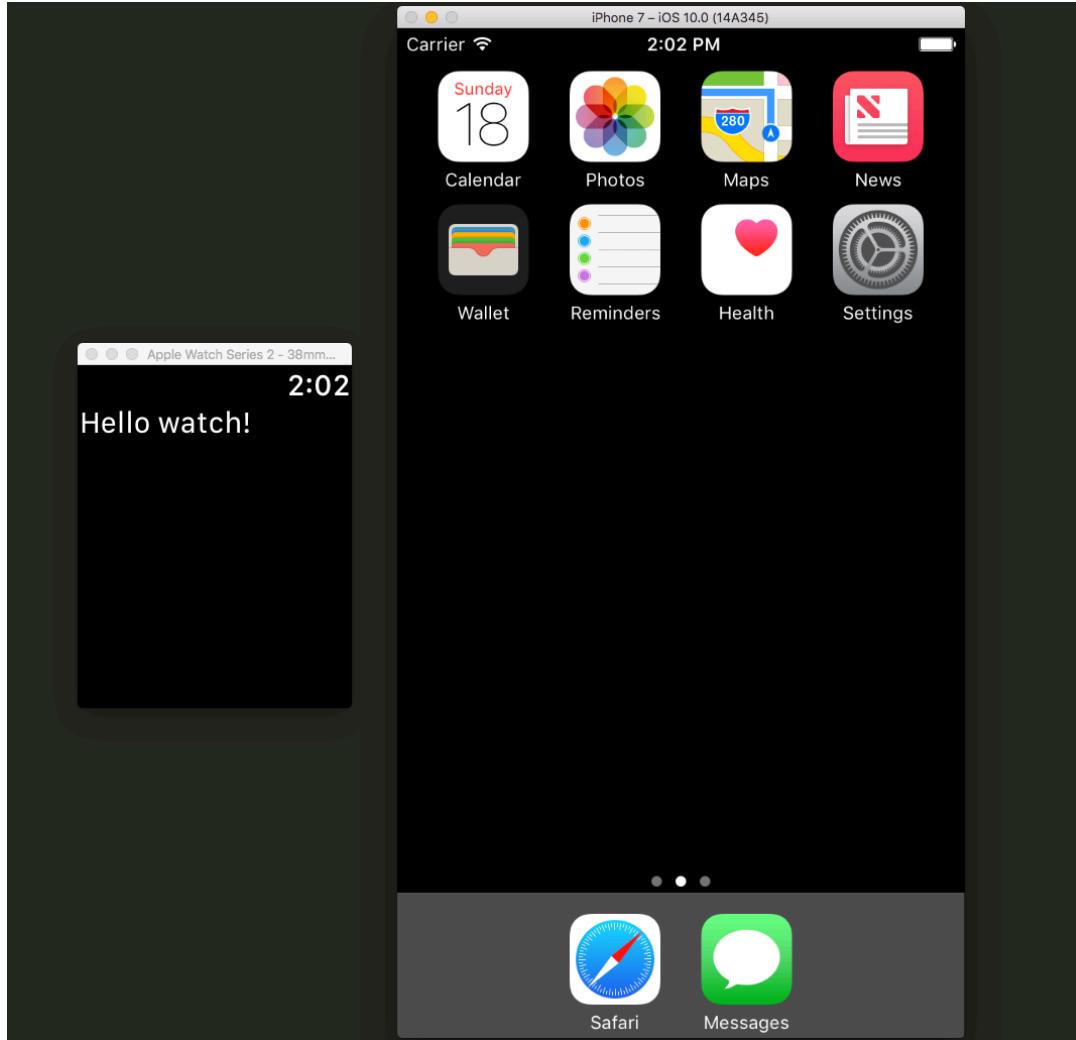
59:59 Timer - Displays a string that counts up or down to a specified time.

Activity Ring Activity Ring - Displays an Activity Ring

Pick a Scheme...and Run!



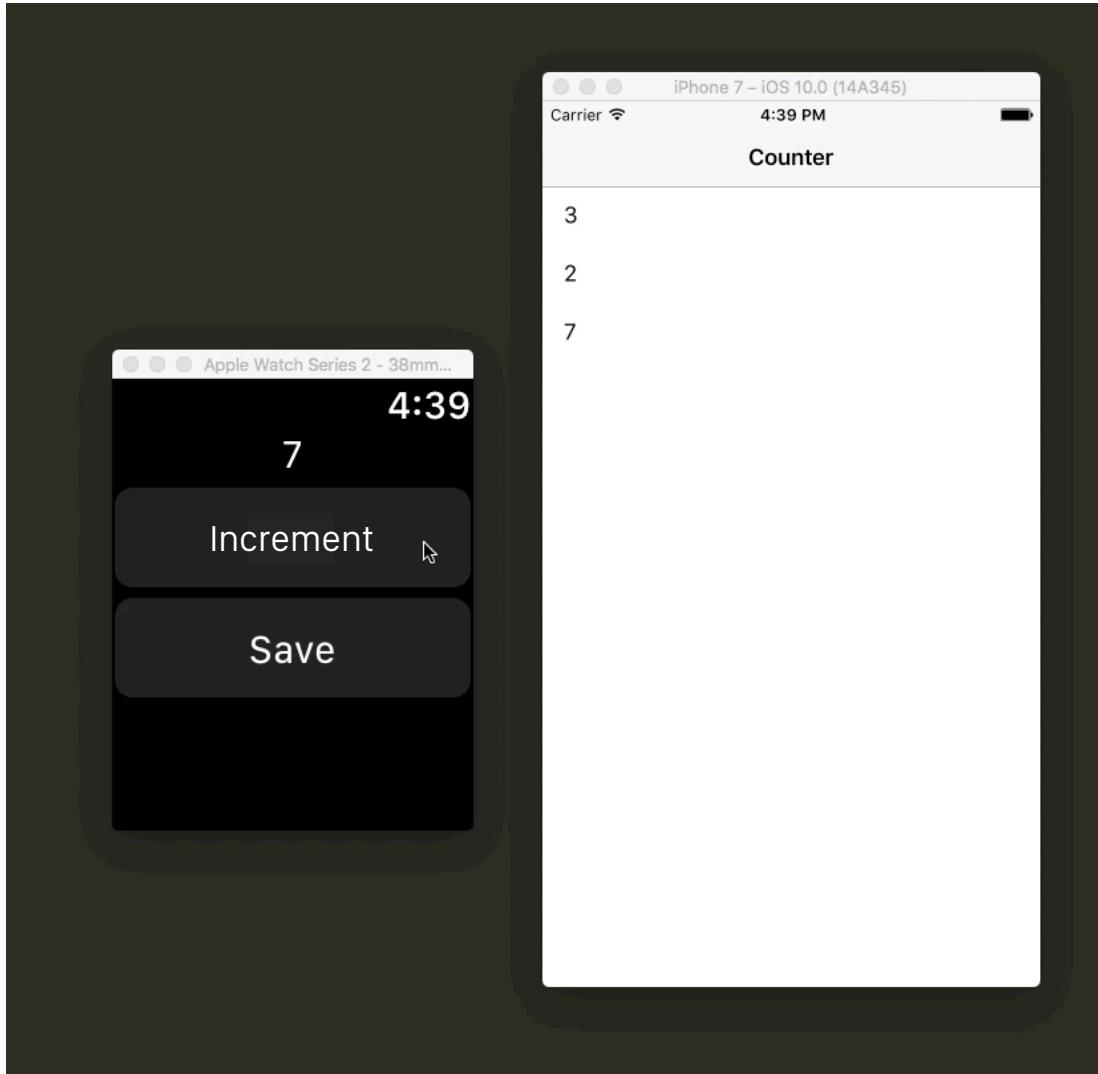
Launch the App!



Watch Counter App (with iPhone Companion App)



Final Product

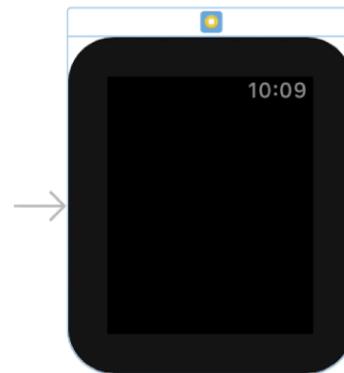


Counter Application

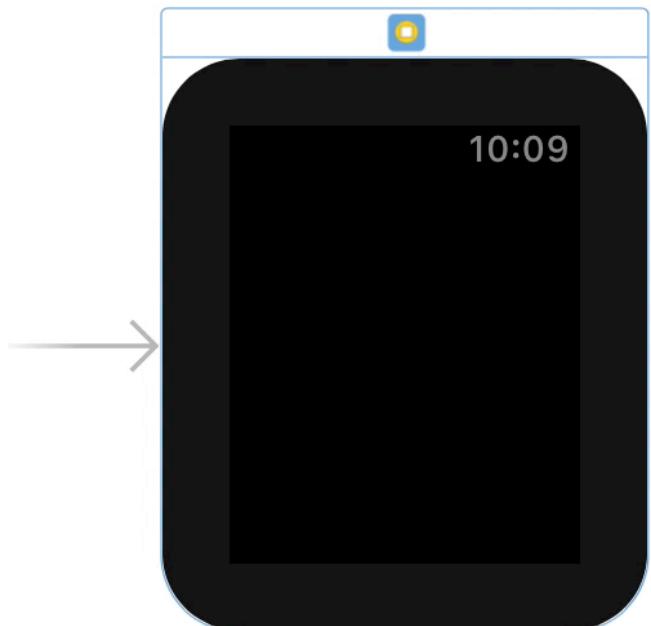
Open the Xcode project that we gave you (Counter.xcodeproj)

<http://bit.ly/2cL7FI4>

On left sidebar, navigate to Counter WatchKit App >
Interface.storyboard



Setup the Counter Label



The Interface Controller settings panel shows the following configuration:

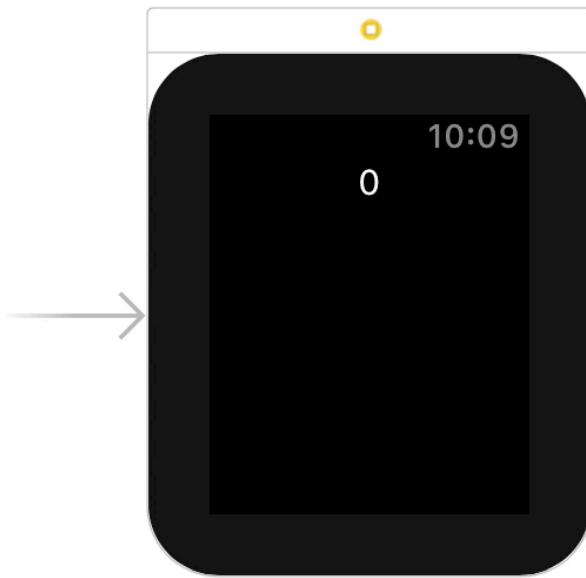
- Identifier: [empty]
- Title: [empty]
- Is Initial Controller:
- Display Activity Indicator When Loading:
- Always Bounce:
- Background: No Image
- Mode: Scale To Fill
- Animate: No
- Color: Default
- Insets: Default
- Spacing: Default

Below the settings panel, there are three cards:

- 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.

Apple Pay - Payment Button - Standard button for initiating Apple Pay transactions.

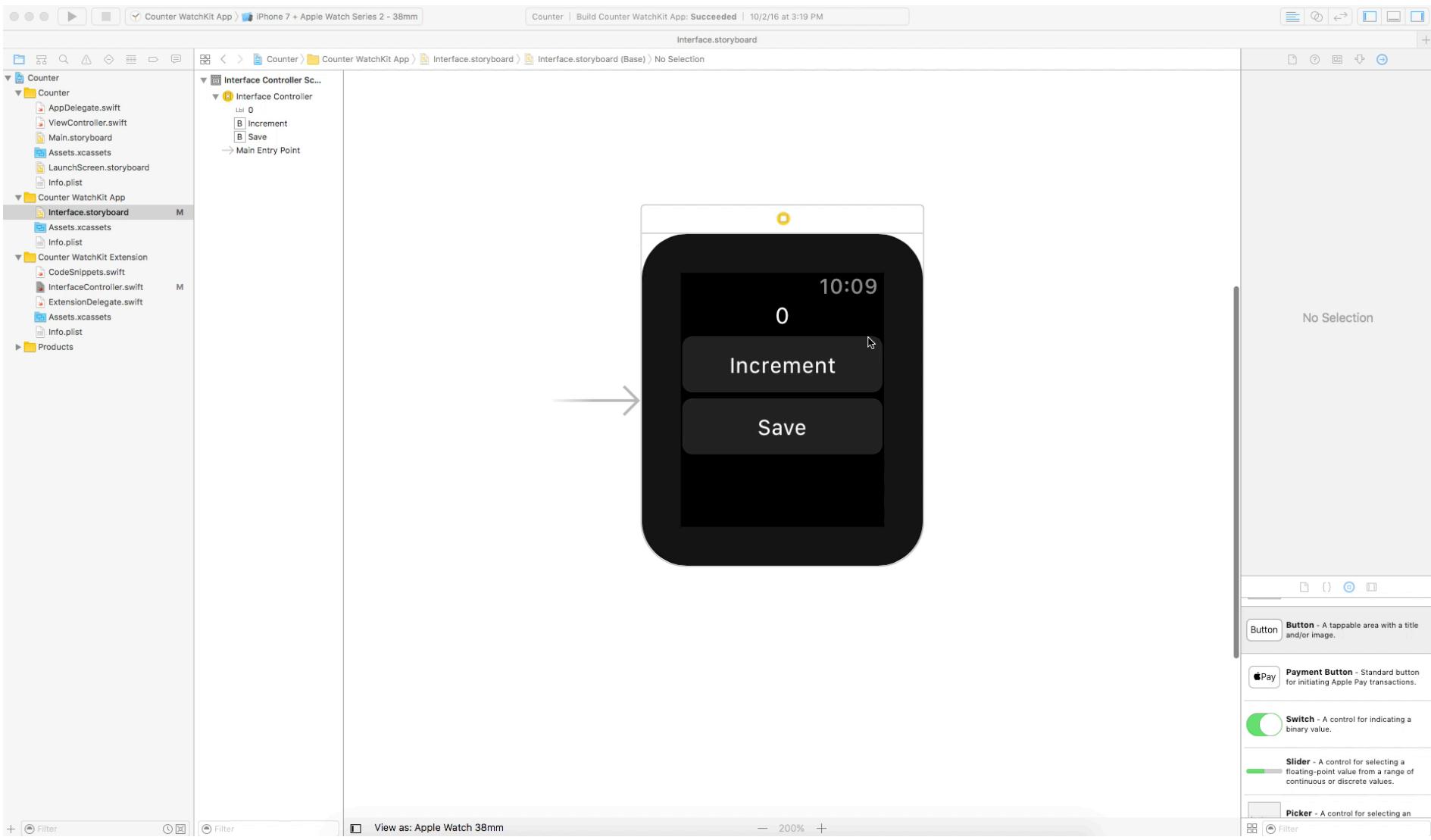
Switch - Switch - A control for indicating a binary value.

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

Picker - 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 image shows a split-screen view of Xcode. On the left, the storyboard interface is displayed, featuring a digital watch face with a black background. The time is set to 10:09. A digital label displays the value '0'. Below the label are two large, rounded rectangular buttons labeled 'Increment' and 'Save'. On the right, the code editor displays the 'InterfaceController.swift' file. The code is a Swift script for a WatchKit extension, starting with several comments explaining the creation and purpose of the file. It includes imports for WatchKit and Foundation, and defines a class 'InterfaceController' that conforms to 'WKInterfaceController'. The class contains sections for setting up connectivity and defining functions for incrementing and saving the counter.

```
// 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 {
    /*
     * 1 - Create a variable for your counter
     */

    /*
     * 4 - Set up Watch Connectivity
     */

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

    /*
     * 6 - Setup your saveCounter() function
     */
}
```

Connect the button actions

The screenshot shows the Xcode interface with two main panes. On the left, the storyboard preview shows a digital watch face with a digital clock at the top displaying "10:09" and a counter below it showing "0". There are two buttons: "Increment" and "Save". A large arrow points from the storyboard preview towards the code editor on the right. The code editor displays the file "InterfaceController.swift" with the following content:

```
// 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
     */

    /*
     * 6 - Setup your saveCounter() function
     */
}
```

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 }
```

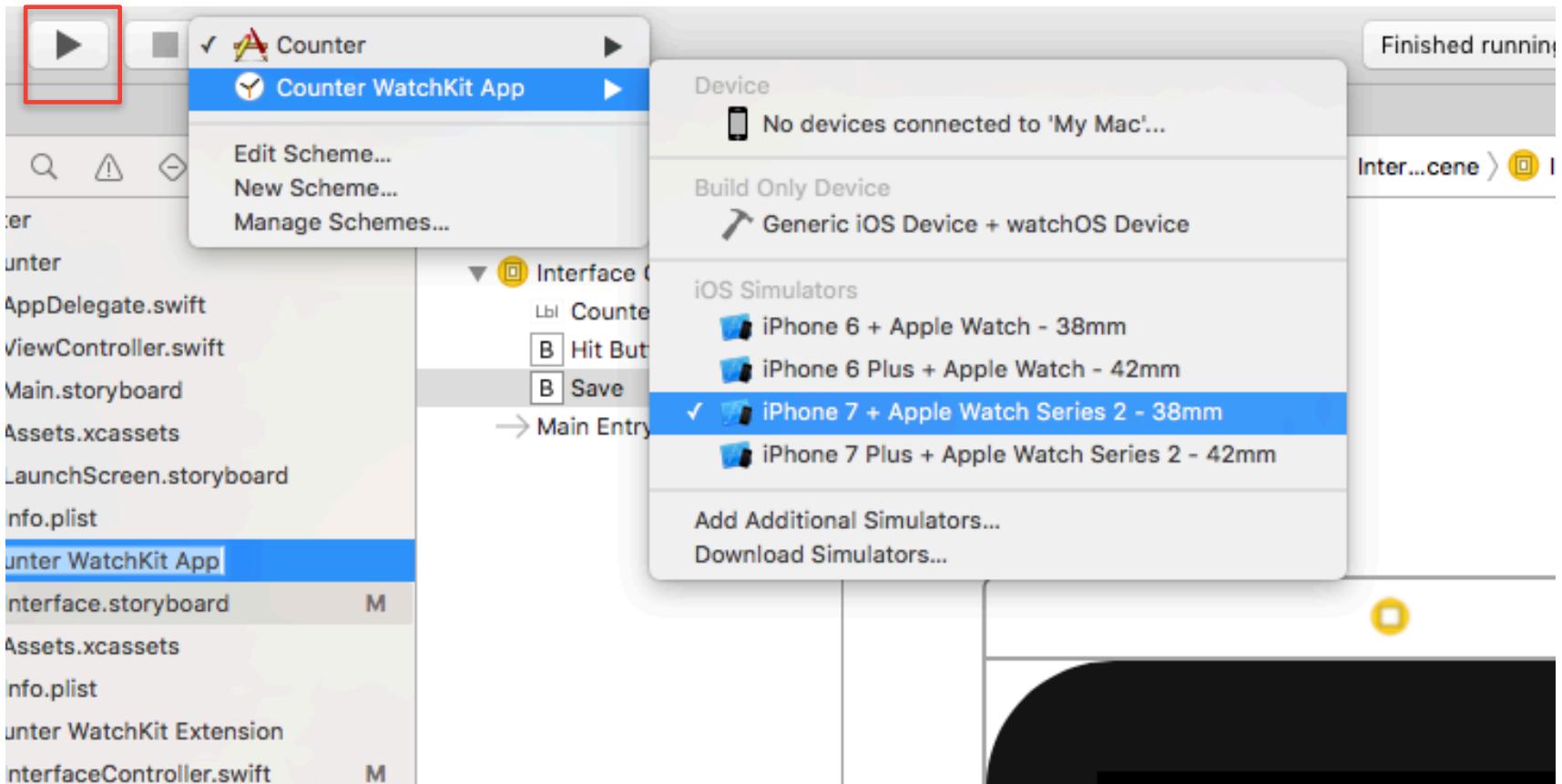
5 – Setup your incrementCounter() function

```
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
```

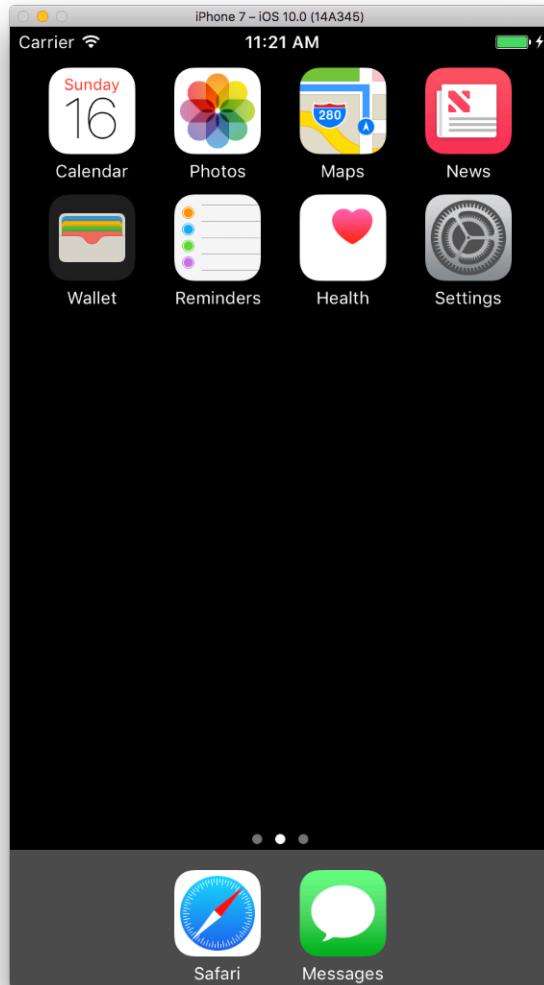
6 – Setup your saveCounter() function

```
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

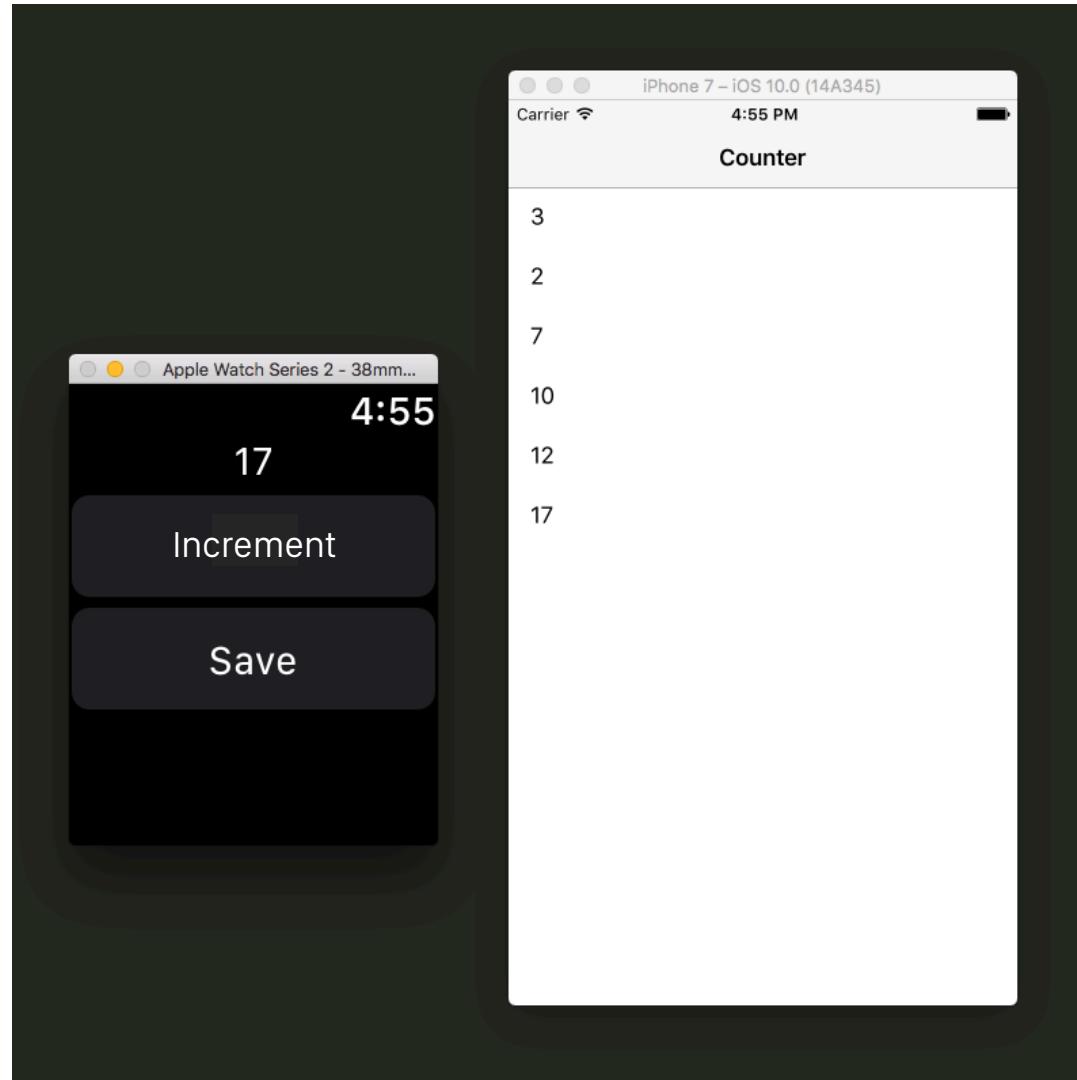


Swipe left to get
to the 2nd page



Click on the
Counter app

Counter Application



Final Project Code

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for 'Pull requests', 'Issues', and 'Gist'. On the right side of the header are icons for notifications, a plus sign, and user profile. Below the header, the repository name 'kristinathai / watchOS3Counter' is displayed, along with buttons for 'Unwatch' (2), 'Unstar' (2), 'Fork' (0), and a profile picture. A secondary navigation bar below the main one includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area starts with a message 'No description or website provided. — Edit'. Below this, there are summary statistics: 12 commits, 1 branch, 0 releases, 2 contributors, and MIT license. There are also buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a prominent green 'Clone or download' button. The main list of commits shows the following entries:

Commit	Message	Time
kristinathai Add code snippets file		Latest commit 924fef7 a minute ago
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/kristinathai/watchOS3Counter>

Resources and Q&A



Kristina's iOS & watchOS Blog

kristina.io

ABOUT · WATCHOS · IOS · PRESENTATIONS & PUBLICATIONS · WRITING · RESUME

WATCHOS

watchOS 3 Key Takeaways from WWDC16 (Part 1)

June 18, 2016 · kristina · Leave a comment

+8 Wow! What a week. I'm really excited about all the new developments in watchOS this year. It's now becoming the platform we all wished we had in the beginning – apps that actually launch, complications as shortcuts, an awesome new dock, etc. 🎉 🌟 ⏳ ❤️ This week definitely passed by in a blur,...

[Continue reading](#)



 [Subscribe for Updates](#)

Subscribe to this blog and receive notifications of new tutorials & posts.

[SUBSCRIBE](#)

Resources

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

Thanks!



Kristina Thai

Kristina_Thai@intuit.com

@kristinathai



Mayuko Inoue

mayuko@patreon.com

@hellomayuko

Thank you

ANITA BORG INSTITUTE
GRACE HOPPER CELEBRATION OF WOMEN IN COMPUTING

Feedback?

@kristinathai @hellomayuko

Rate and review the session on our mobile app

Download at <http://bit.ly/ghc16app>
or search GHC 16 in the app store