Apple Watch - Exploring the possibilities

Antoine Barrault
@devasapps

Cocoaheads - Campinas 05/05/2015

WatchKit Framework

The old friends renamed

- WKInterfaceController
- WKInterfaceObject,WKInterfaceButton,WKInterfaceImage,WKInterfaceLabel, WKInterfaceSwitch
- WKInterfaceTable
- WKMapView
- WKInterfaceDevice

NEW CHALLENGER APPROACHING !

Our new challengers

- WKInterfaceSeparator
- WKInterfaceDate
- WKInterfaceTimer
- WKInterfaceSlider

WKInterfaceGroup

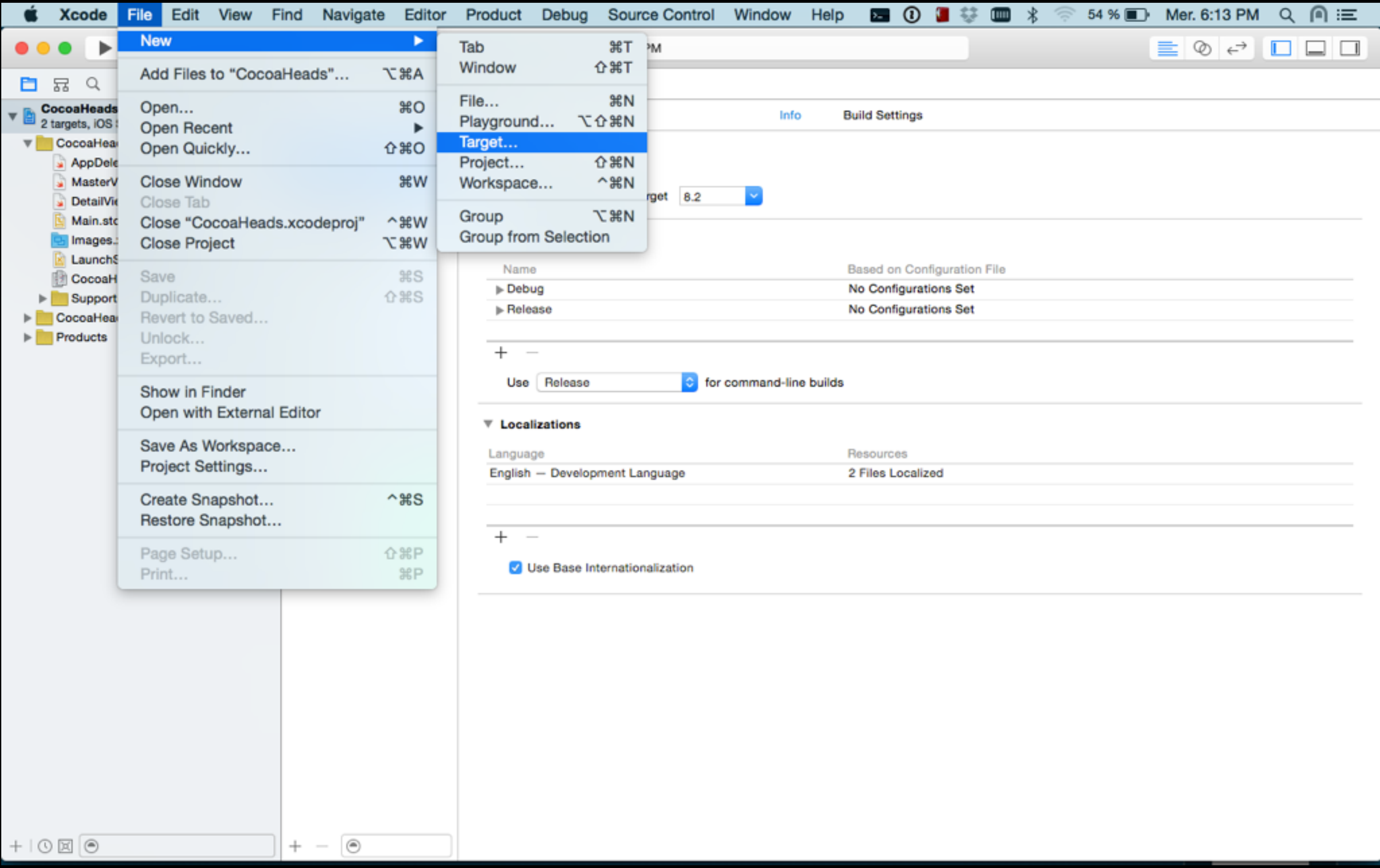
- Container of other elements
- Manage layout (horizontal or vertical)
- Can have a background image
- Can set a background color

WKInterfaceMenu

- Call on force press
- Add using storyboard or programmatically
- Up to 4 buttons
- One action for each button

Let’s begin

Add an apple watch target to your iOS app



8.2

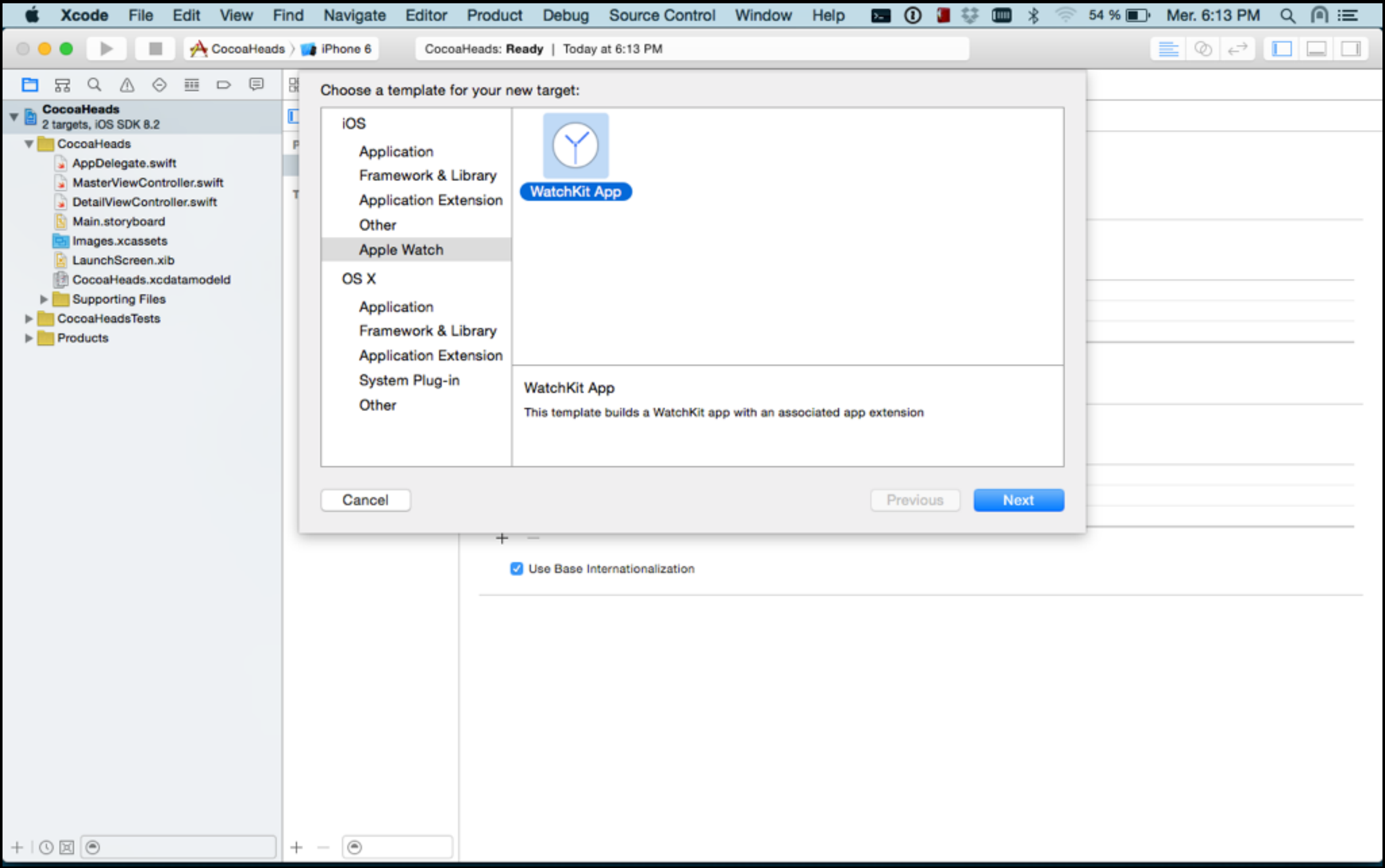
Name	Based on Configuration File
▶ Debug	No Configurations Set
▶ Release	No Configurations Set

Use Release for command-line builds

Localizations

Language	Resources
English — Development Language	2 Files Localized

☑ Use Base Internationalization

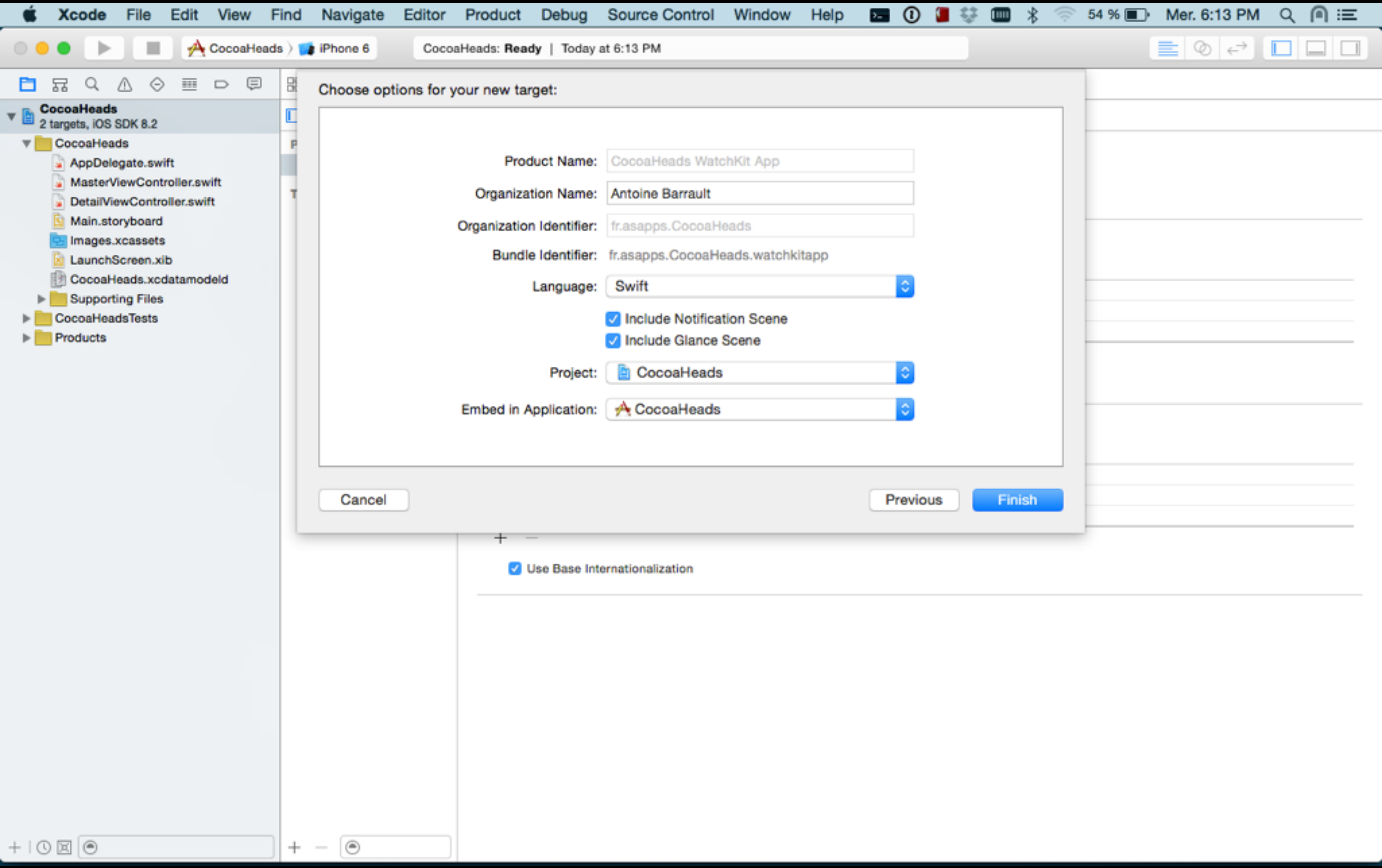


Choose a template for your new target:

WatchKit App

This template builds a WatchKit app with an associated app extension

☑ Use Base Internationalization



Choose options for your new target:

Product Name: CocoaHeads WatchKit App

Organization Name: Antoine Barrault

Organization Identifier: fr.asapps.CocoaHeads

Bundle Identifier: fr.asapps.CocoaHeads.watchkitapp

Language: Swift

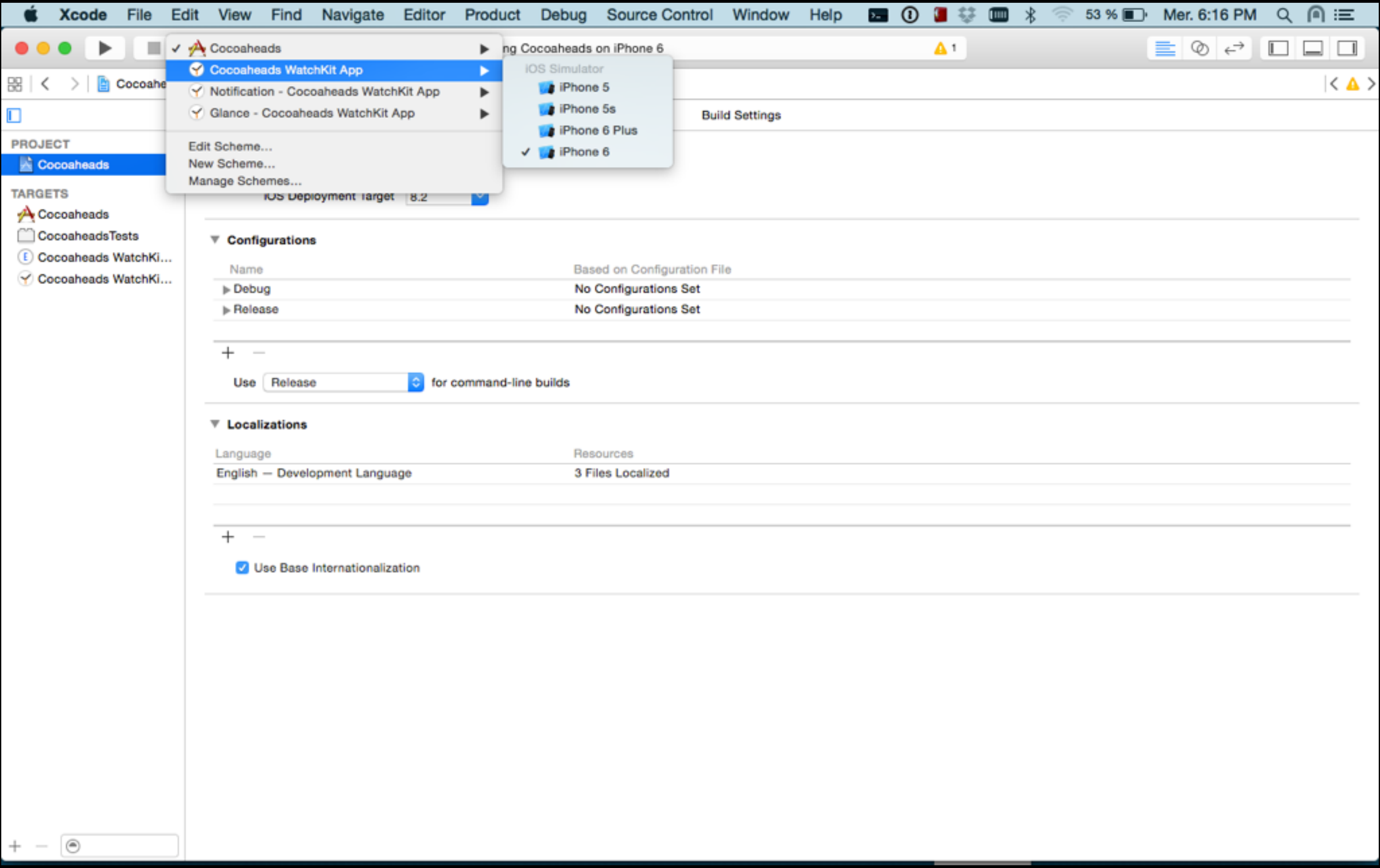
☑ Include Notification Scene

☑ Include Glance Scene

Project: CocoaHeads

Embed in Application: CocoaHeads

☑ Use Base Internationalization



iOS Deployment Target: 8.2

Configurations

Name	Based on Configuration File
▶ Debug	No Configurations Set
▶ Release	No Configurations Set

+ —

Use Release for command-line builds

Localizations

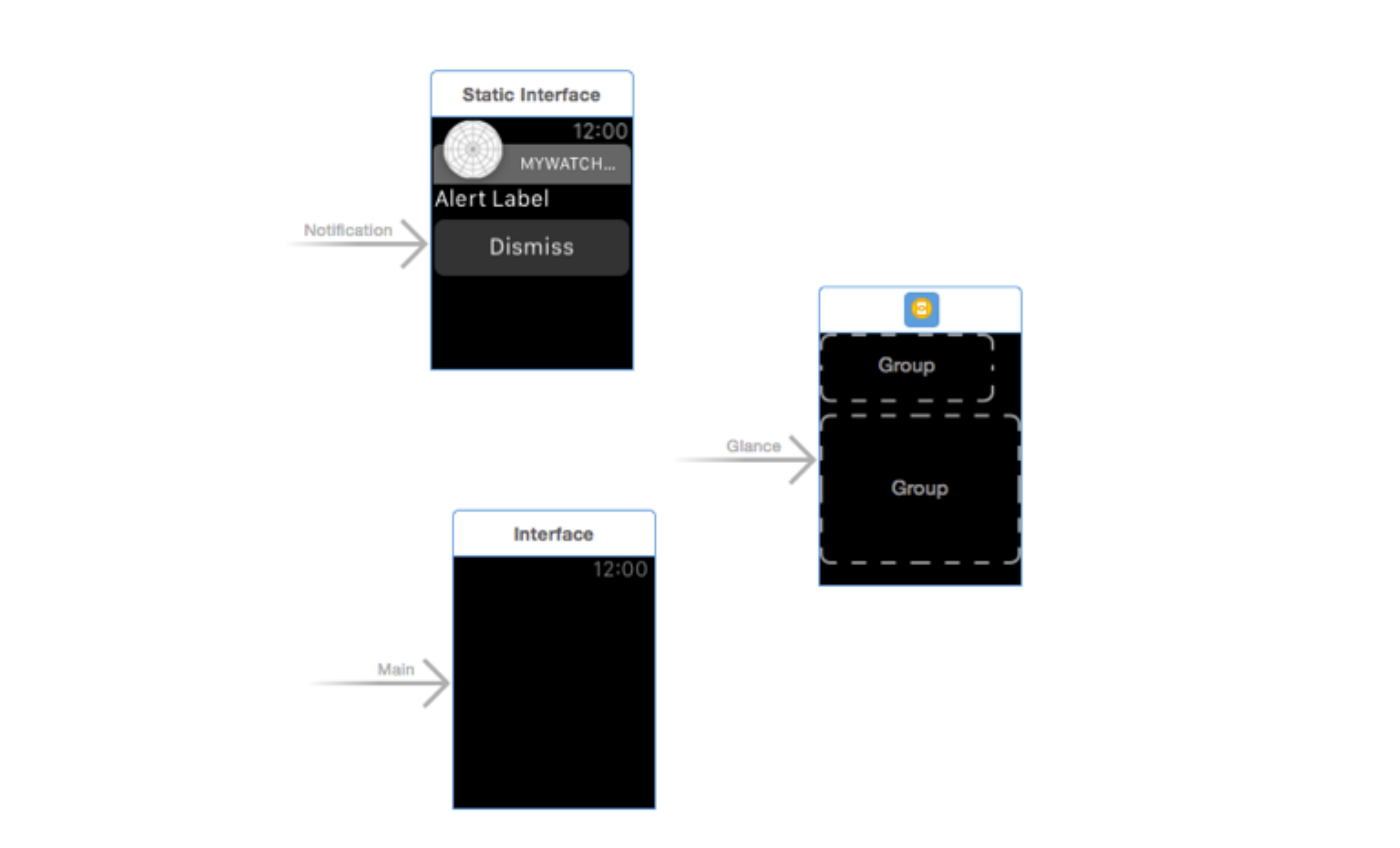
Language	Resources
English — Development Language	3 Files Localized

+ —

☑ Use Base Internationalization

Demo

It’s all about storyboard



One storyboard to rules them all

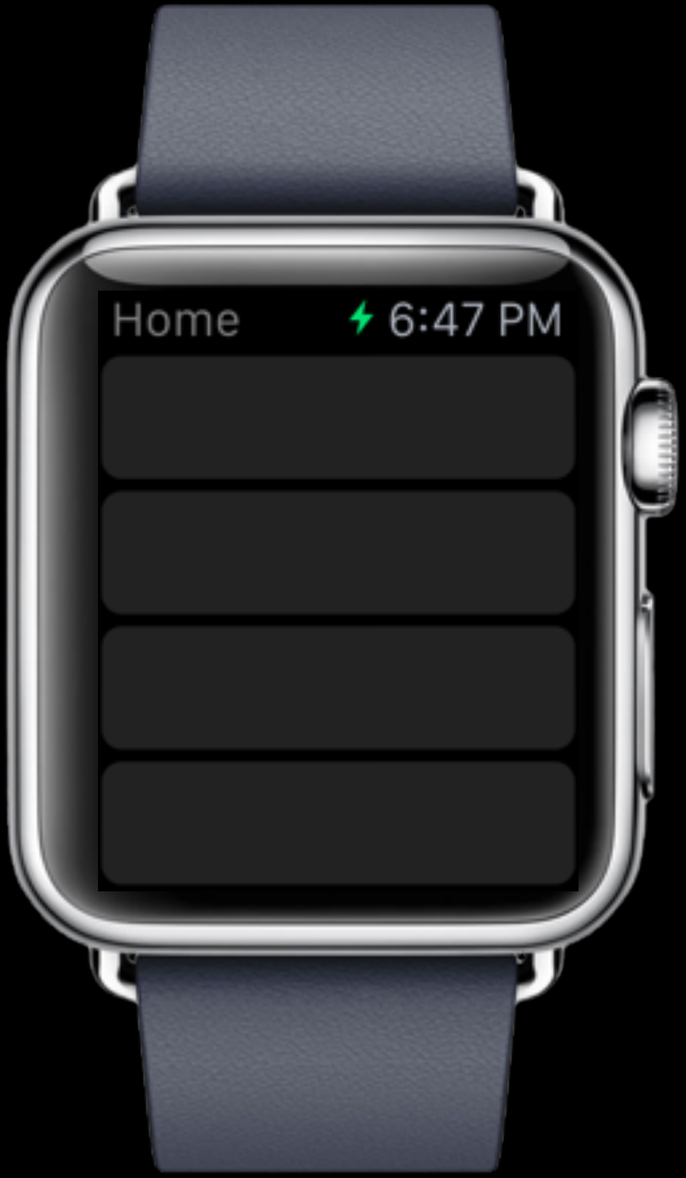
It’s all about storyboard

WKInterfaceTable

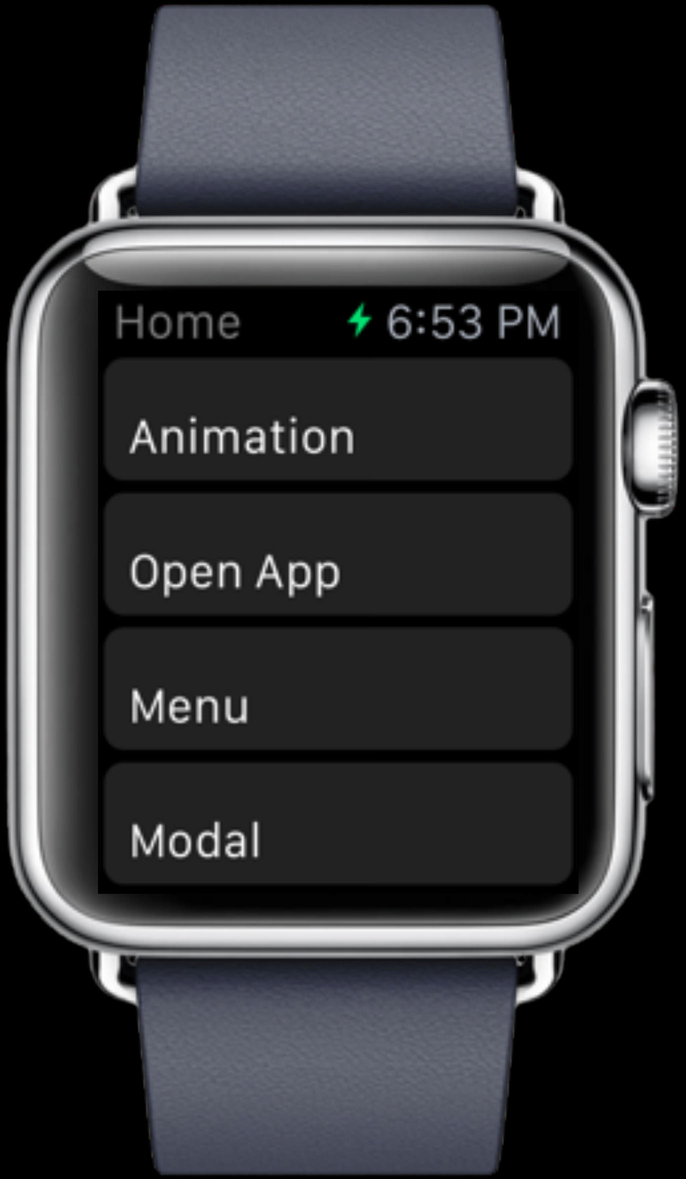
Create a WKInterfaceTable

- Add a WKInterfaceTable to your storyboard
- Set number of row
- No delegate, no datasource

Done!



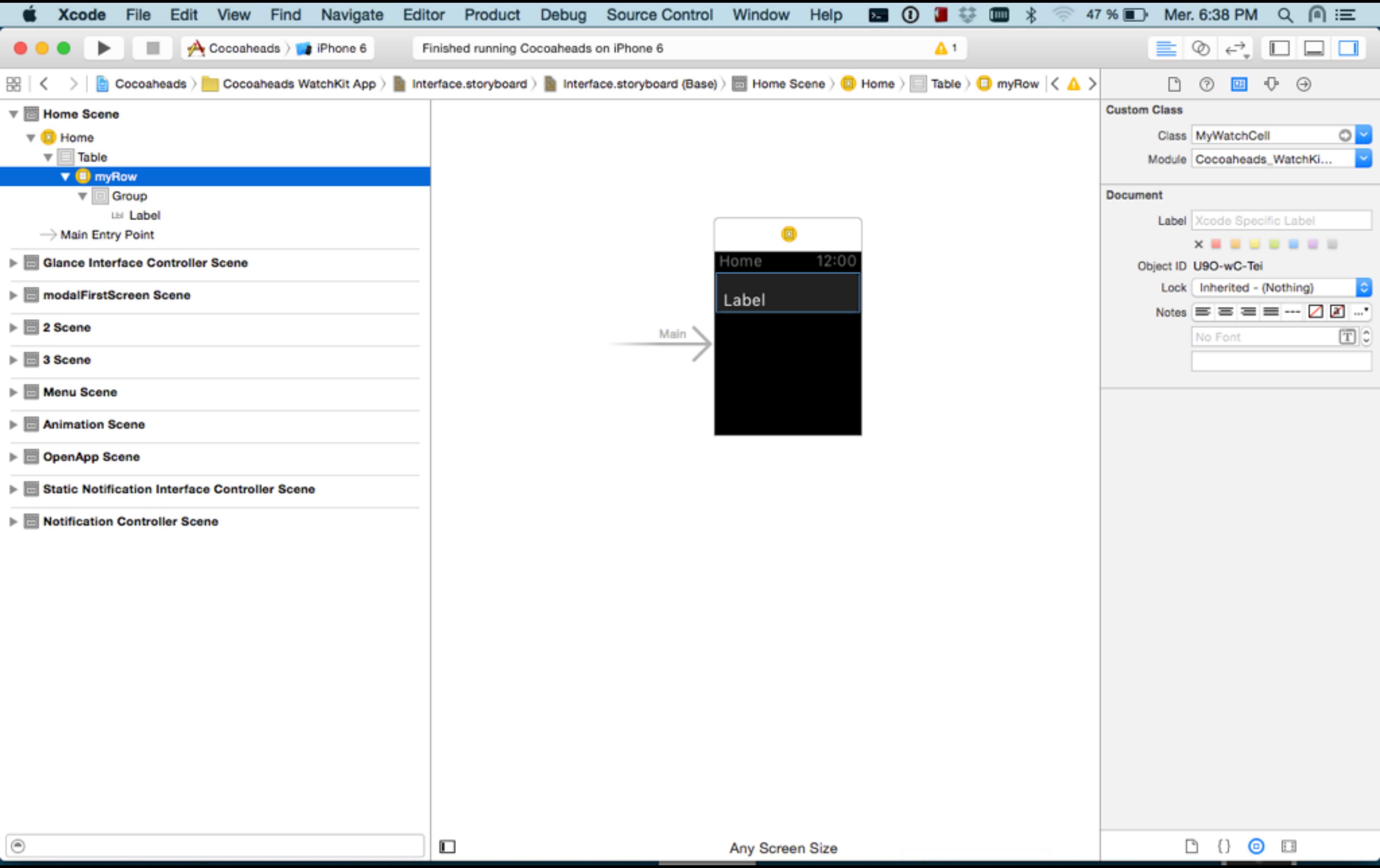
What about our menu?



Home ⚡ 6:53 PM

Create a custom cell

```swift
import WatchKit
class MyWatchCell: NSObject {
    @IBOutlet weak var labelOnCell: WKInterfaceLabel!
}
```



Finished running Cocoaheads on iPhone 6

Custom Class

Class MyWatchCell

Module Cocoaheads_WatchKi...

Document

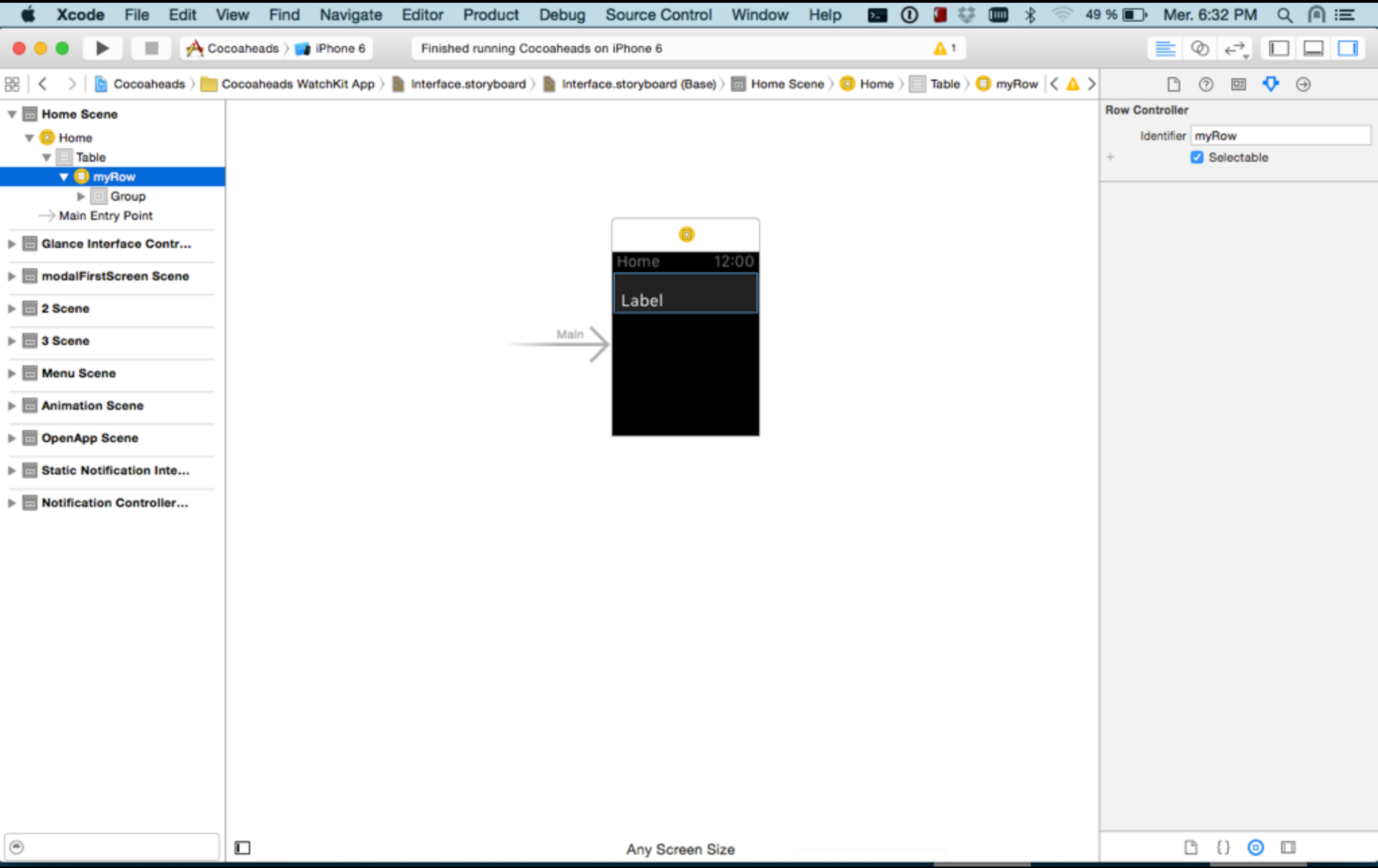
Label Xcode Specific Label

Object ID U9O-wC-Tei

Lock Inherited - (Nothing)

Notes

No Font


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
49 % Mer. 6:32 PM
Cocoaheads 〉 iPhone 6
Finished running Cocoaheads on iPhone 6
⚠ 1
Cocoaheads 〉 Cocoaheads WatchKit App 〉 Interface.storyboard 〉 Interface.storyboard (Base) 〉 Home Scene 〉 Home 〉 Table 〉 myRow
Home Scene
Home
Table
myRow
Group
Main Entry Point
Glance Interface Contr...
modalFirstScreen Scene
2 Scene
3 Scene
Menu Scene
Animation Scene
OpenApp Scene
Static Notification Inte...
Notification Controller...
Home
12:00
Label
Main
Any Screen Size

Row Controller

Identifier myRow

☑ Selectable

Configure each row

```
@IBOutlet weak var table: WKInterfaceTable!

var rows = ["Animation","Open App","Menu","Modal"]

override func willActivate() {
   super.willActivate()

   table.setNumberOfRows(rows.count, withRowType: "myRow")

   for (index, value) in enumerate(rows) {

     let atualRowForThisIndex =
     table.rowControllerAtIndex(index) as MyWatchCell
     atualRowForThisIndex.labelOnCell.setText(value)

   }
}
```

Let start animation!

Let start animation!

- ~~UIView.animateWithDuration~~
- ~~setFrame, setOrigin~~
- Only one solution : WKInterfaceImage

```
func startAnimatingWithImagesInRange(imageRange: NSRange,
duration: NSTimeInterval, repeatCount: Int)
```

Let start animation!

```
@IBOutlet weak var image: WKInterfaceImage!

override func willActivate() {

  image.setImageNamed("frame_")
  let animateRange = NSRange(location: 0, length: 75)
  image.startAnimatingWithImagesInRange(animateRange,
         duration: 4.0, repeatCount: 0)
  super.willActivate()

}
```

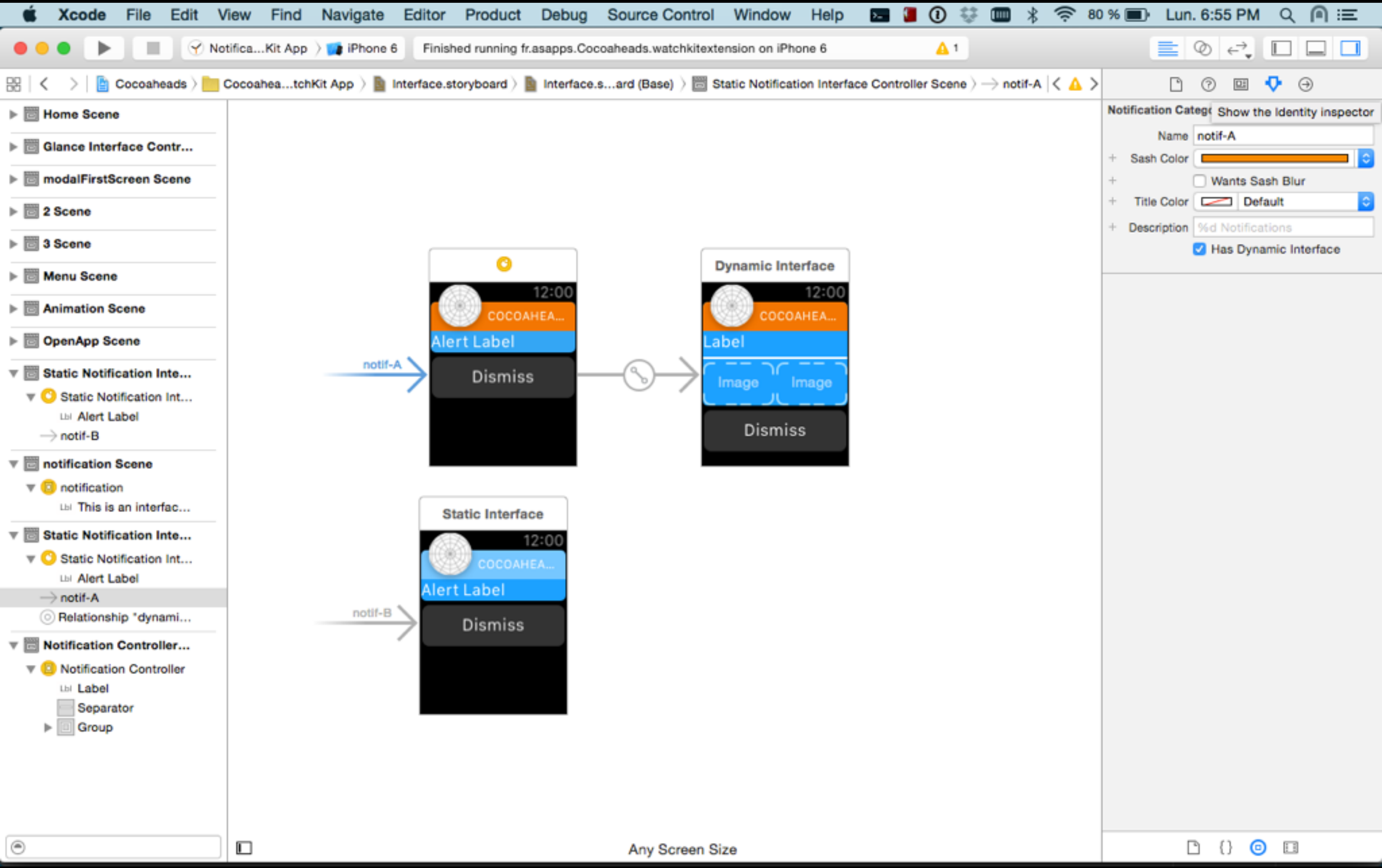
Communicate with the iPhone app

The simplest way

```
let userInfo = ["method":"glance"]

WKInterfaceController.openParentApplication(userInfo, reply: {

(theReply : [NSObject : AnyObject]!, error : NSError!) -> Void in

    if error == nil {

      if let dico = theReply {

         if let number  = dico["eventsNumber"] as? NSNumber {

                self.numberLabel.setText("\(number)")

           }
       }

    }


 })
```

```
func application(application: UIApplication!, handleWatchKitExtensionRequest
        userInfo: [NSObject : AnyObject]!,
        reply: (([NSObject : AnyObject]!) -> Void)!) {

        if let method  = userInfo["method"] as? String {
            var number : Int?
            switch method {
            case "checkNumber" :  number = numberOfEvents()
            case "addEvent":  number = addEvent()
            case "deleteEvents":  deleteAll()
            default :  number = addEvent()
            }
            let dico = ["eventsNumber":NSNumber(integer: number!)]
            reply(dico)
        }
    }
```

Watch Notifications



Finished running fr.asapps.Cocoaheads.watchkitextension on iPhone 6

Notification Category

Name notif-A

Sash Color

□ Wants Sash Blur

Title Color Default

Description %d Notifications

☑ Has Dynamic Interface

Dynamic interface

- Subclass WKUserNotificationInterfaceController
- Implement :

```
func didReceiveRemoteNotification(
remoteNotification: [NSObject : AnyObject], withCompletion
 completionHandler: ((WKUserNotificationInterfaceType) -> Void))

{
   //do initialization of your notification interface here
   completionHandler(.Custom)

}
```

Handle notification interaction

- Implement handleActionWithIdentifier on the main entry point interface of the apple watch app for foreground notification

```
func handleActionWithIdentifier(identifier: String?,
forRemoteNotification remoteNotification: [NSObject : AnyObject]) {

    //do some stuff with the notification
    self.presentControllerWithName("notification", context: nil)

}
```

Test the notifications

```
{
    "aps": {
        "alert": {
            "body": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
            "title": "Optional title"
        },
        "category": "notif-A"
    },
    "WatchKit Simulator Actions": [
                                   {
                                   "title": "First Button",
                                   "identifier": "firstButtonAction"
                                   },
                                   {
                                   "title": "Second Button",
                                   "identifier": "secondButtonAction"
                                   },
                                   ]
}
```

Register notification actions

- WatchKit simulator actions are just for testing
- Need to register your notification actions and categories
- The same as iOS 8 interactive notifications

Register notification actions

```
func registerSettingsAndCategories() {

     var categories = NSMutableSet()

     var firstAction = UIMutableUserNotificationAction()
     firstAction.title = "First Action"
     firstAction.identifier = "firstButtonAction"
     firstAction.activationMode = UIUserNotificationActivationMode.Background
     firstAction.authenticationRequired = false

     var secondAction = UIMutableUserNotificationAction()
     secondAction.title = "Second Action"
     secondAction.identifier = "secondButtonAction"
     secondAction.activationMode = UIUserNotificationActivationMode.Foreground
     secondAction.authenticationRequired = false

     var notifACategory = UIMutableUserNotificationCategory()
     notifACategory.setActions([firstAction, secondAction],
         forContext: UIUserNotificationActionContext.Default)
     notifACategory.identifier = "notif-A"


     categories.addObject(notifACategory)

     var settings = UIUserNotificationSettings(forTypes: (.Alert | .Badge | .Sound),
         categories: categories as Set<NSObject>)

     UIApplication.sharedApplication().registerUserNotificationSettings(settings)
 }
```

Text Input

Text Input

```
self.presentTextInputControllerWithSuggestions(["YES","NO","MAYBE"],
    allowedInputMode: .AllowAnimatedEmoji) { (answer : [AnyObject]!) -> Void


    println(answer)

}
```

```
enum WKTextInputMode : Int {

    case Plain // text (no emoji) from dictation + suggestions
    case AllowEmoji // text plus non-animated emoji from dictation +
suggestions
    case AllowAnimatedEmoji // all text, animated emoji (GIF data)
}
```

Thanks.

Antoine Barrault - iOS Developer at VivaReal
Twitter : @devasapps
barrault01@gmail.com