Apple Watch - Exploring the possibilities

Antoine Barrault
@devasapps

Cocoaheads - SP 25/03/2015

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

WKUserNotificationInterface Controller

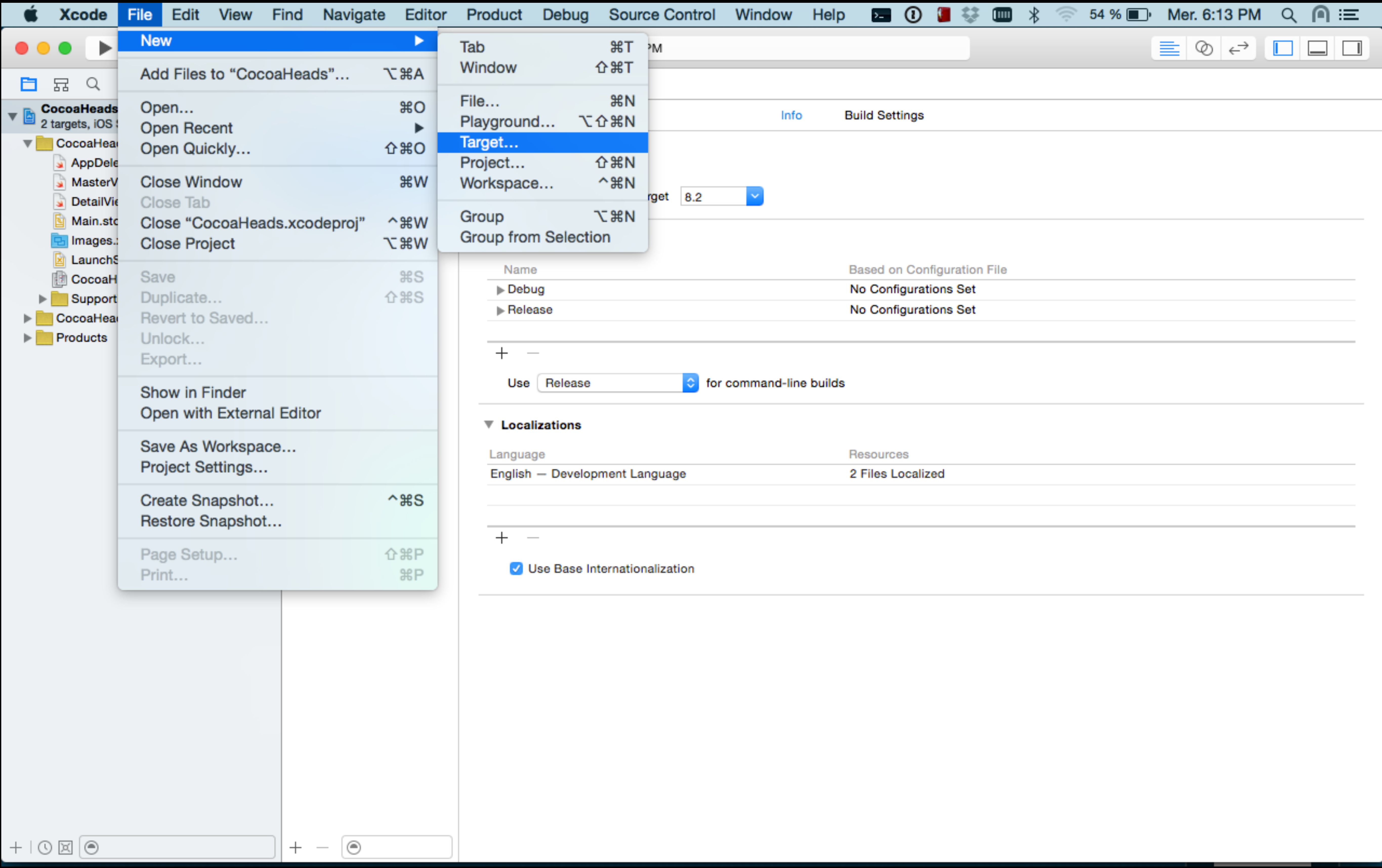
- Class that manages local and remote notifications

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

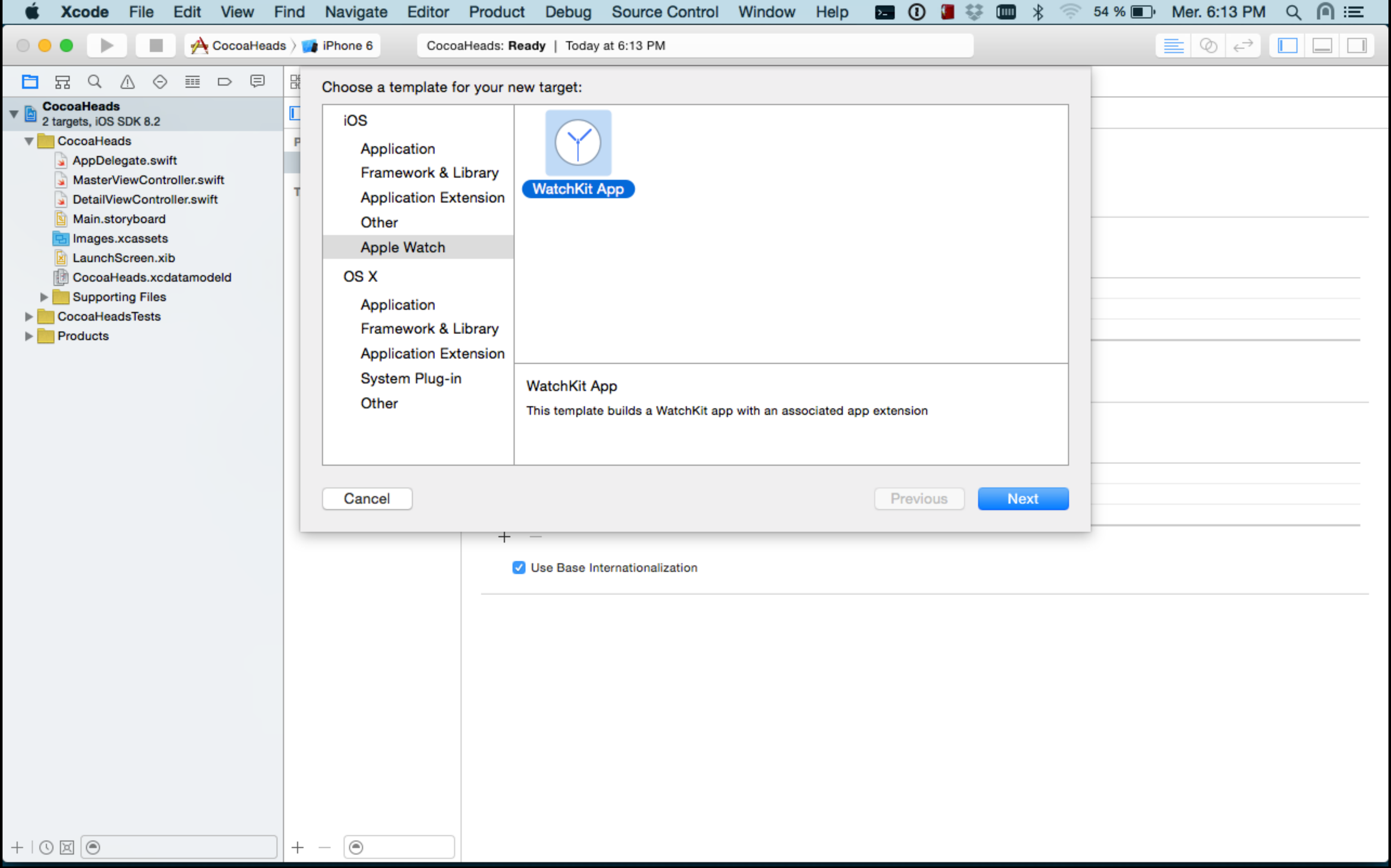


Name	Based on Configuration File
▶ Debug	No Configurations Set
▶ Release	No Configurations Set

Use Release for command-line builds

Language	Resources
English — Development Language	2 Files Localized

☑ Use Base Internationalization



Choose a template for your new target:

iOS

- Application
- Framework & Library
- Application Extension
- Other
- Apple Watch

OS X

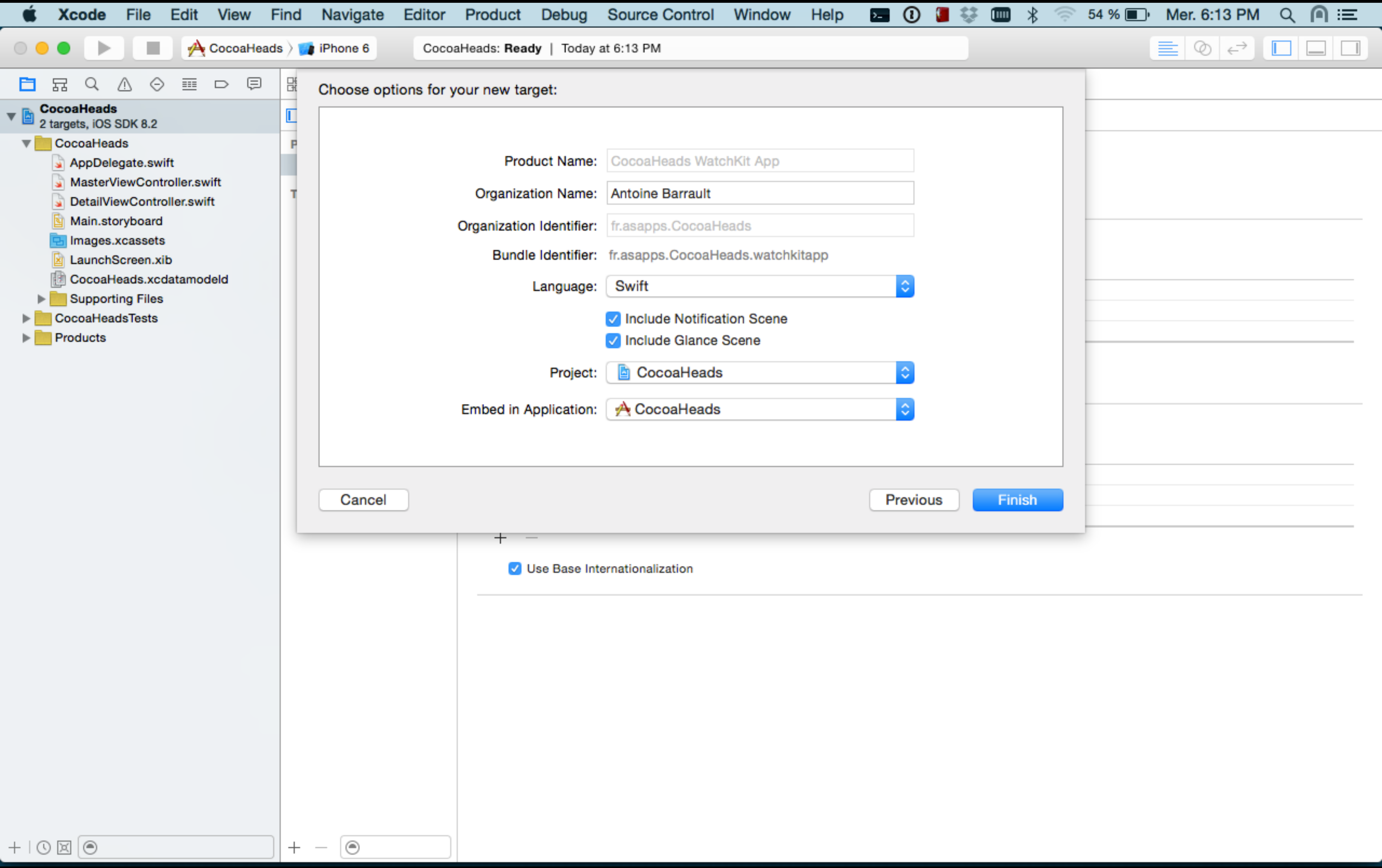
- Application
- Framework & Library
- Application Extension
- System Plug-in
- Other

WatchKit App

This template builds a WatchKit app with an associated app extension

Cancel | Previous | Next

☑ 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

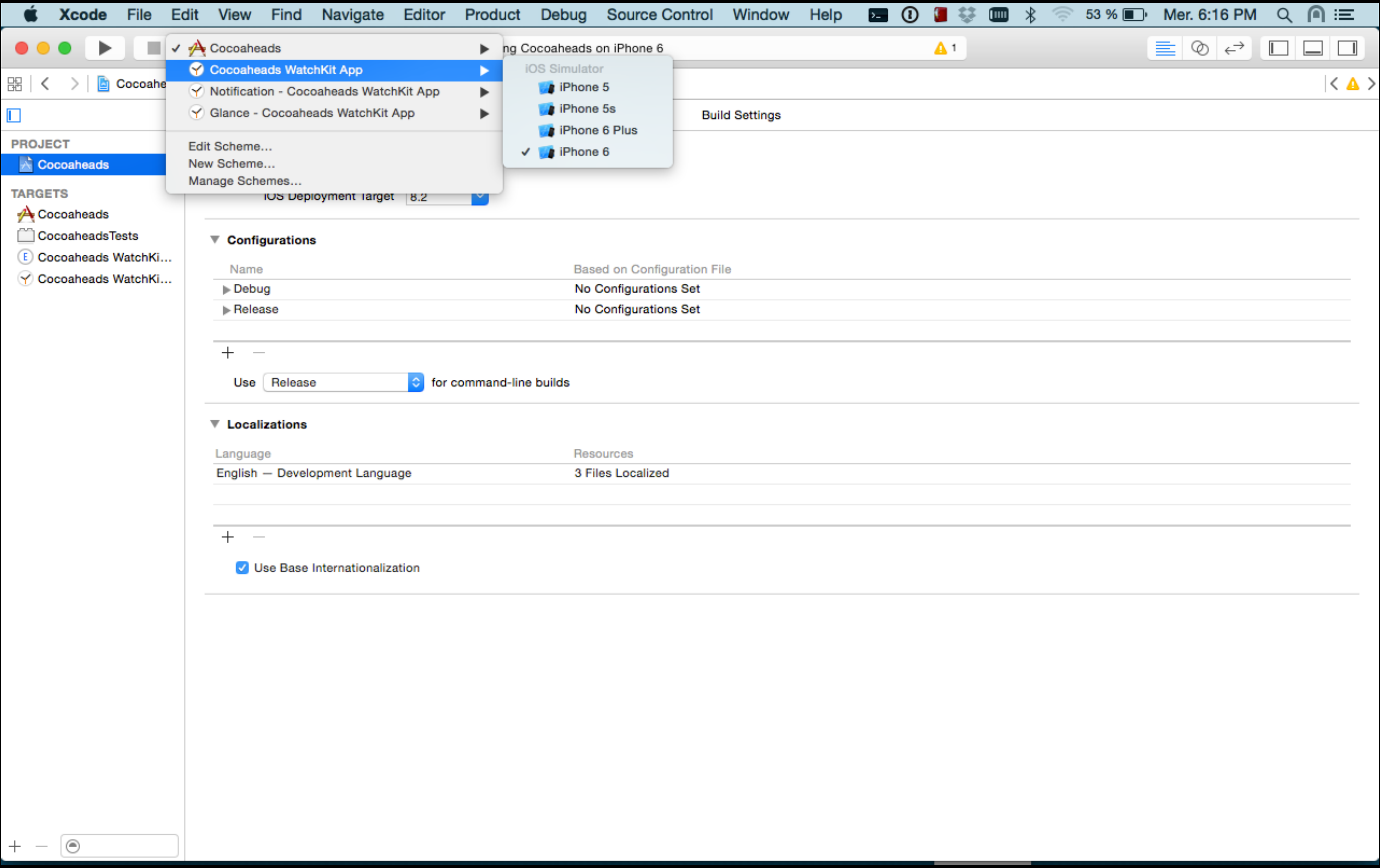
☑ Include Notification Scene

☑ 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

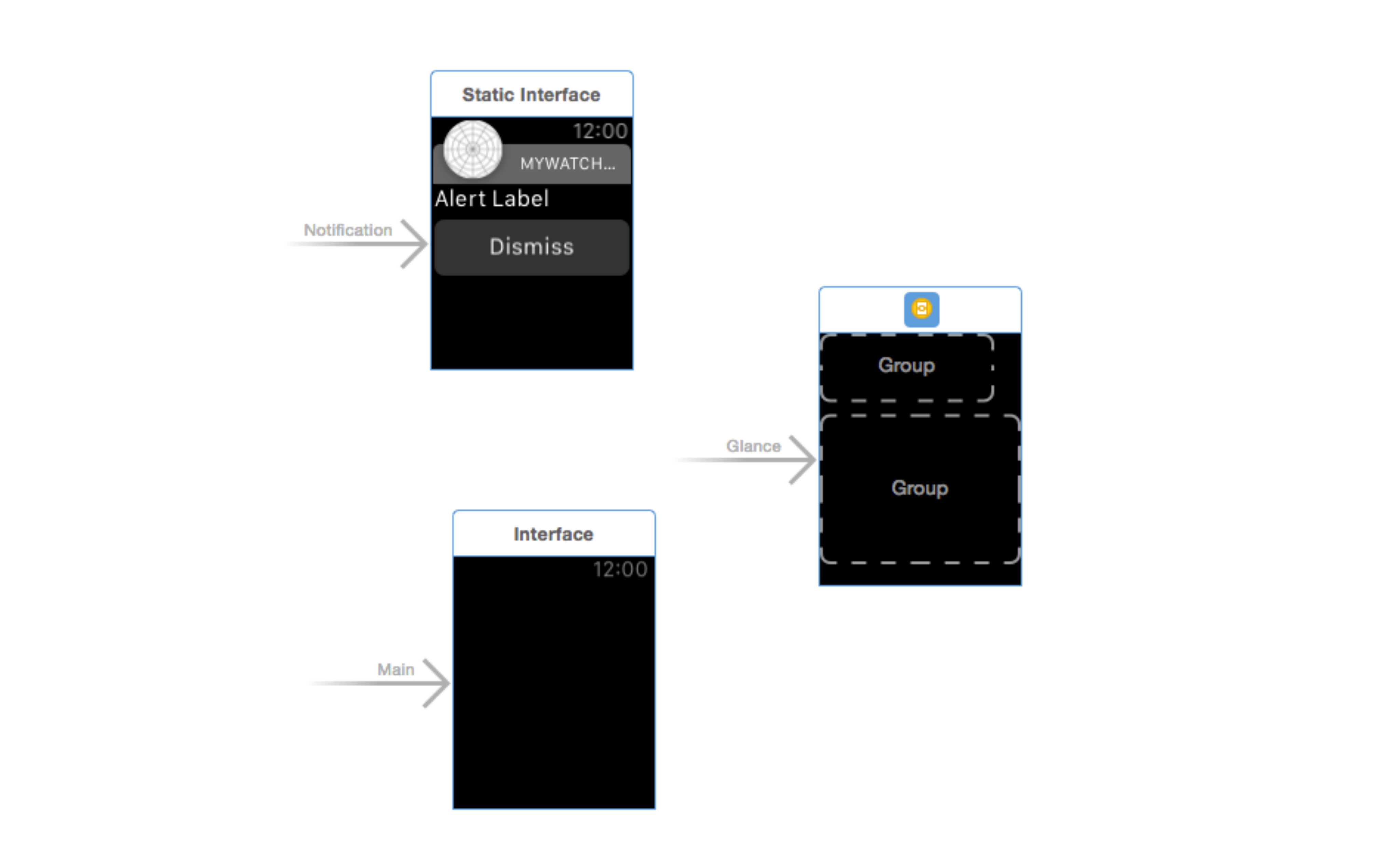
Localizations

Language	Resources
English — Development Language	3 Files Localized

☑ Use Base Internationalization

Demo

It’s all about storyboard



One storyboard to rules them all

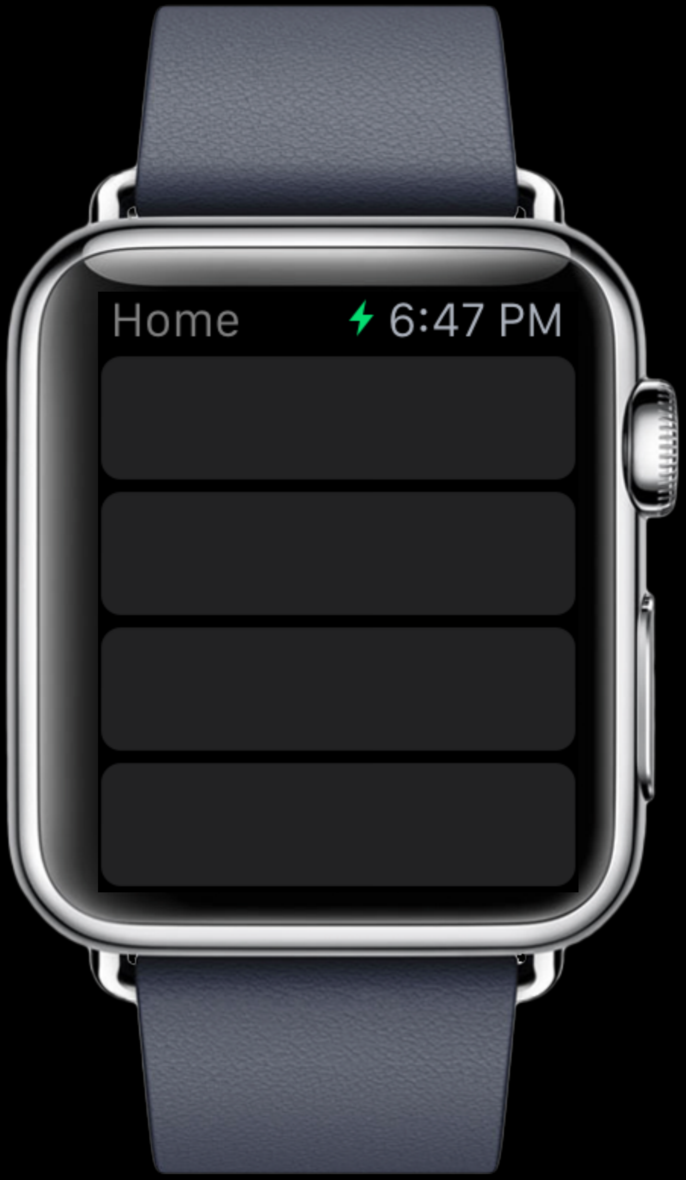
It’s all about storyboard

WKInterfaceTable

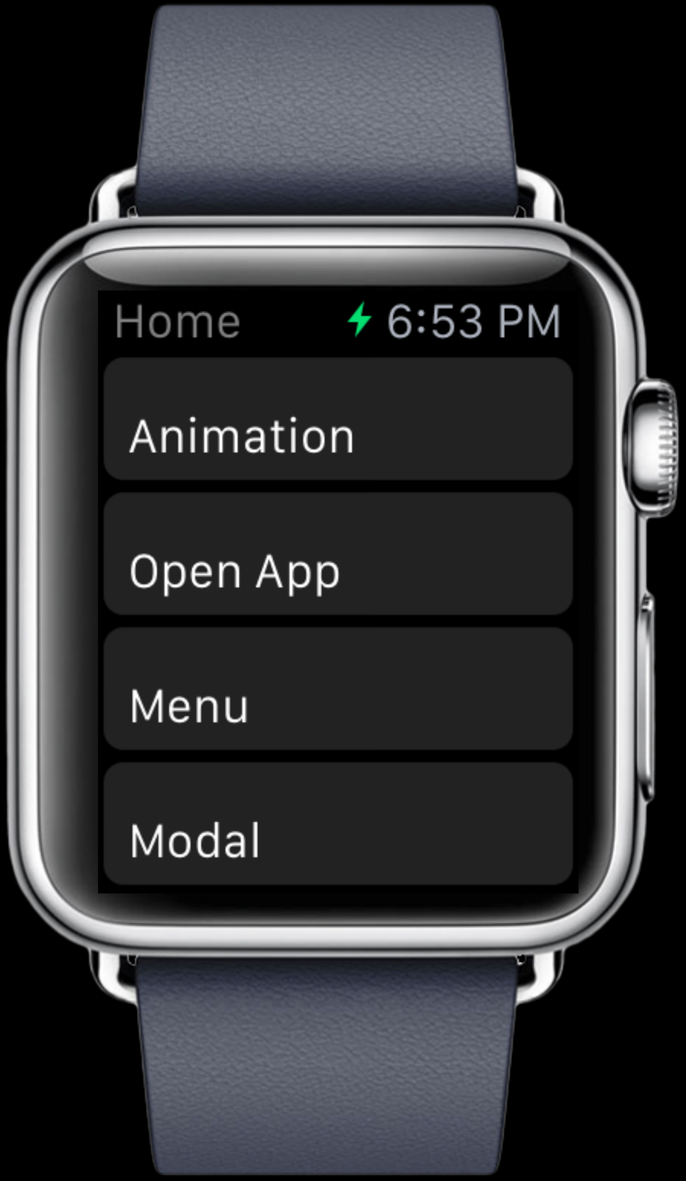
Create a WKInterfaceTable

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

Done!



What about our menu?



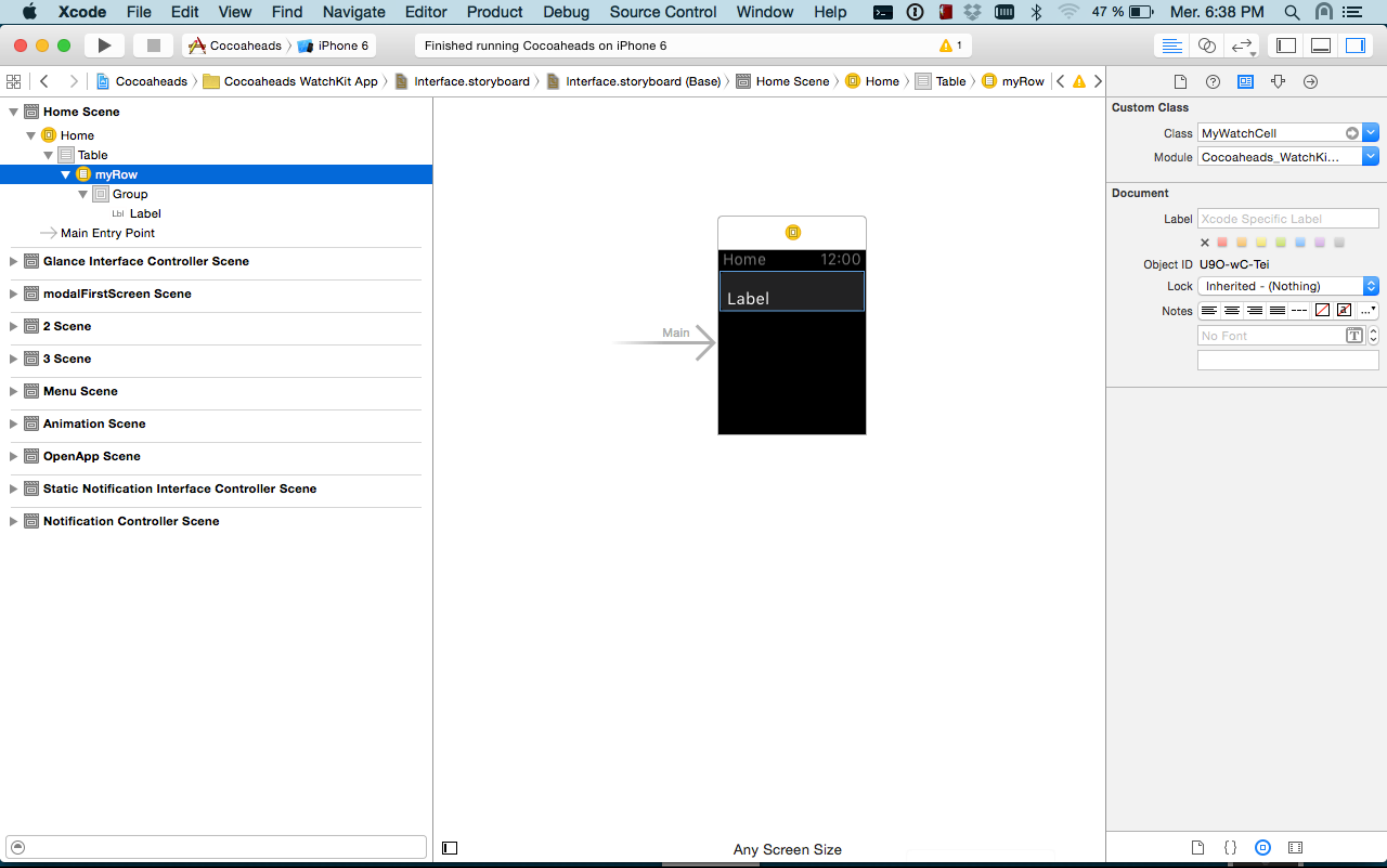
Create a custom cell

```
import WatchKit

class MyWatchCell: NSObject {

    @IBOutlet weak var labelOnCell: WKInterfaceLabel!

}
```



Finished running Cocoaheads on iPhone 6

Cocoaheads › Cocoaheads WatchKit App › Interface.storyboard › Interface.storyboard (Base) › Home Scene › Home › Table › myRow

```
▼ Home Scene
  ▼ Home
    ▼ Table
      ▼ myRow
        ▼ Group
            Lbl Label
    → Main Entry Point
▶ Glance Interface Controller Scene
▶ modalFirstScreen Scene
▶ 2 Scene
▶ 3 Scene
▶ Menu Scene
▶ Animation Scene
▶ OpenApp Scene
▶ Static Notification Interface Controller Scene
▶ Notification Controller Scene
```

Custom Class

Class: MyWatchCell

Module: Cocoaheads_WatchKi...

Document

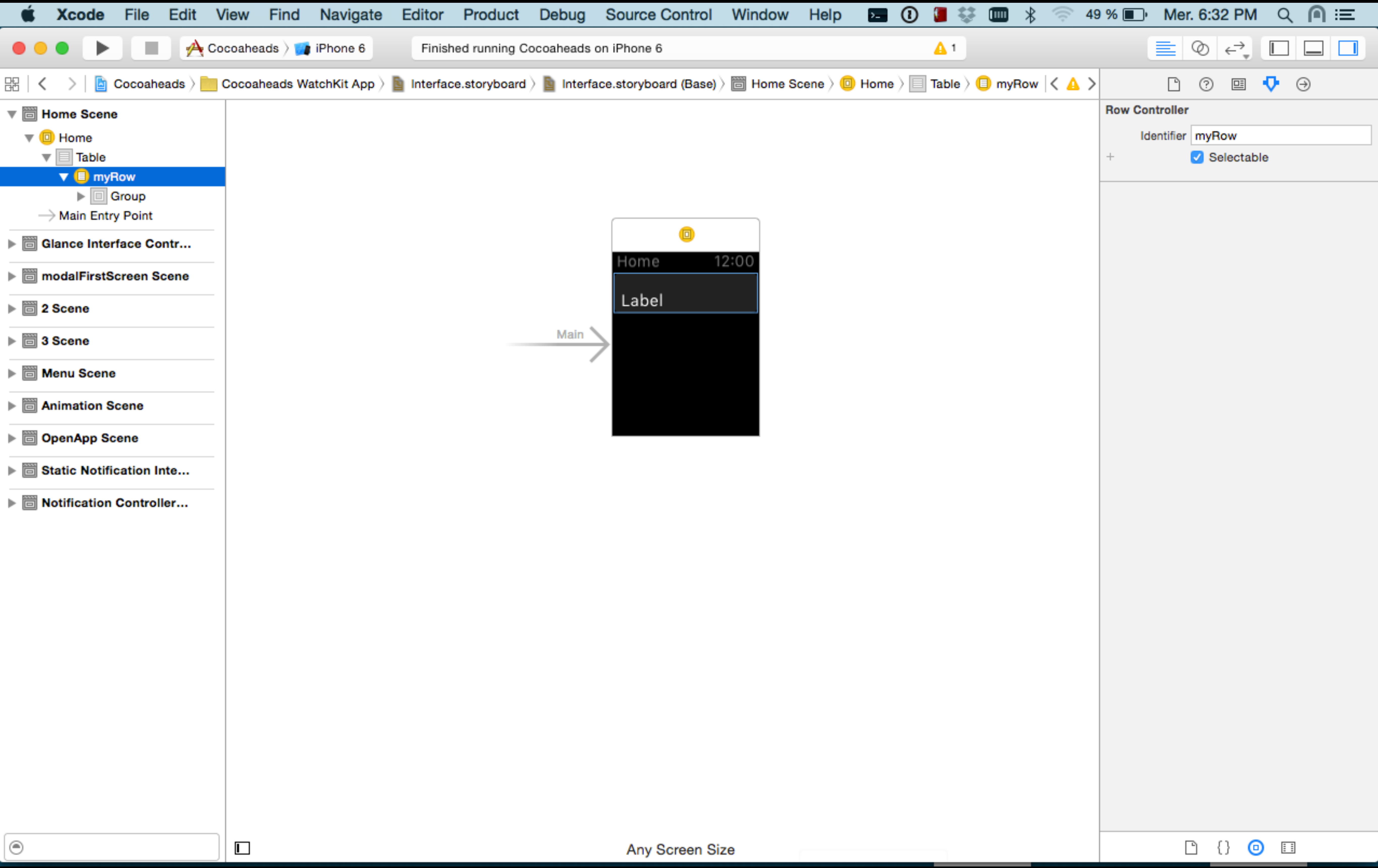
Label: Xcode Specific Label

Object ID: U9O-wC-Tei

Lock: Inherited - (Nothing)

Notes

No Font



Finished running Cocoaheads on iPhone 6

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)
        }
    }
```

What's more

- Text input
- Received notifications on the watch

Thanks.

Antoine Barrault - iOS Developer at VivaReal
Twitter : @devasapps
antoine.barrault@vivareal.com

VivaReal