

EXTENSIONS IN IOS 8

(like peanut butter and chocolate)

BEN SCHEIRMAN

@subdigital
benscheirman.com

A wide-angle photograph of the Houston skyline at sunset. The sky is a mix of orange, pink, and blue, with scattered clouds. Numerous skyscrapers and buildings are visible, many with their windows lit up. The foreground shows a dense urban landscape with various building styles. The overall tone is warm and vibrant.

CHAIONE

Houston, TX
@chaione

NSSCREENCAST



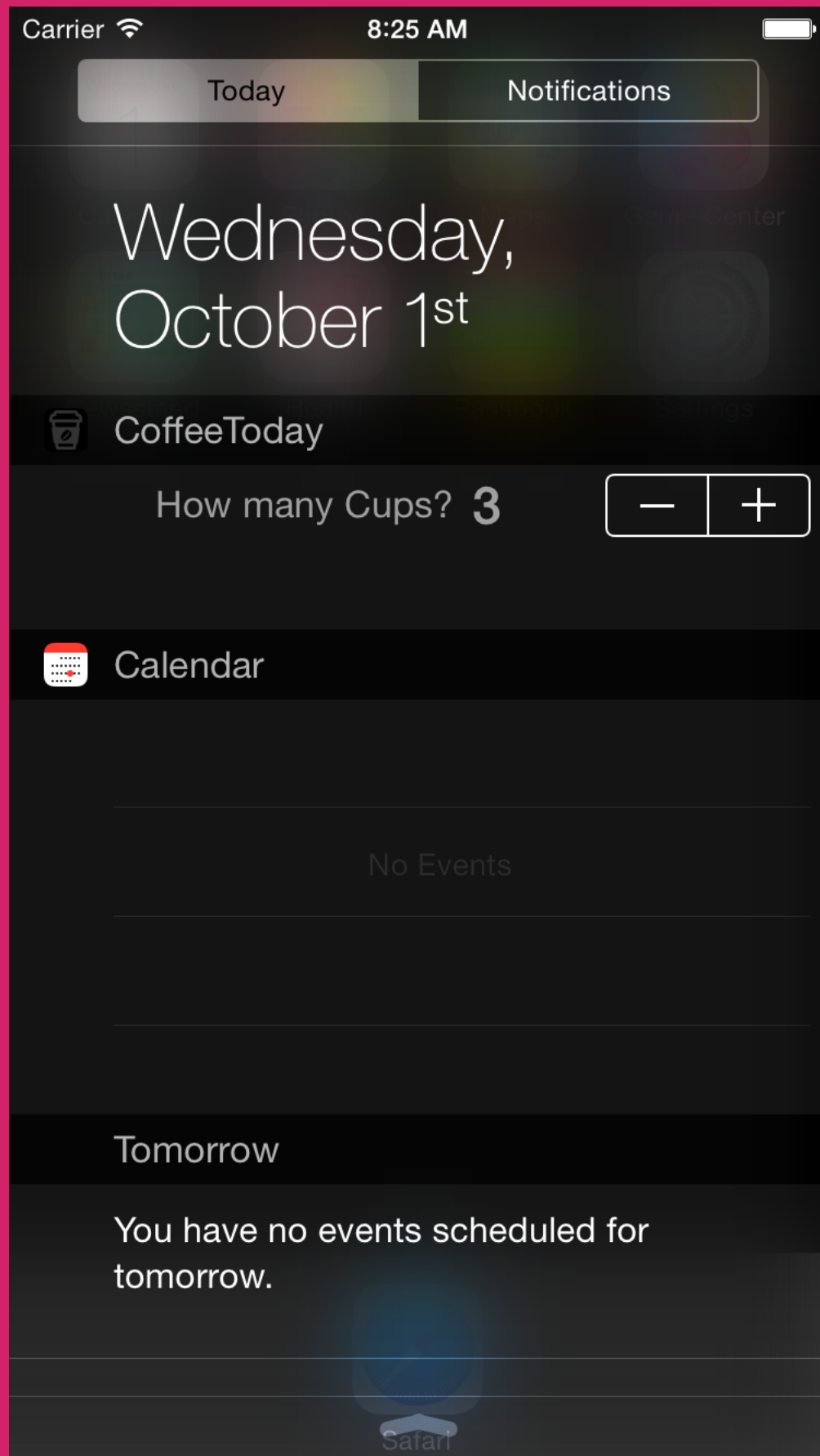
EXTENSIONS!



YOUR APP + OTHER APPS

YOUR APP +  'S APPS

EXTENSION TYPES AVAILABLE

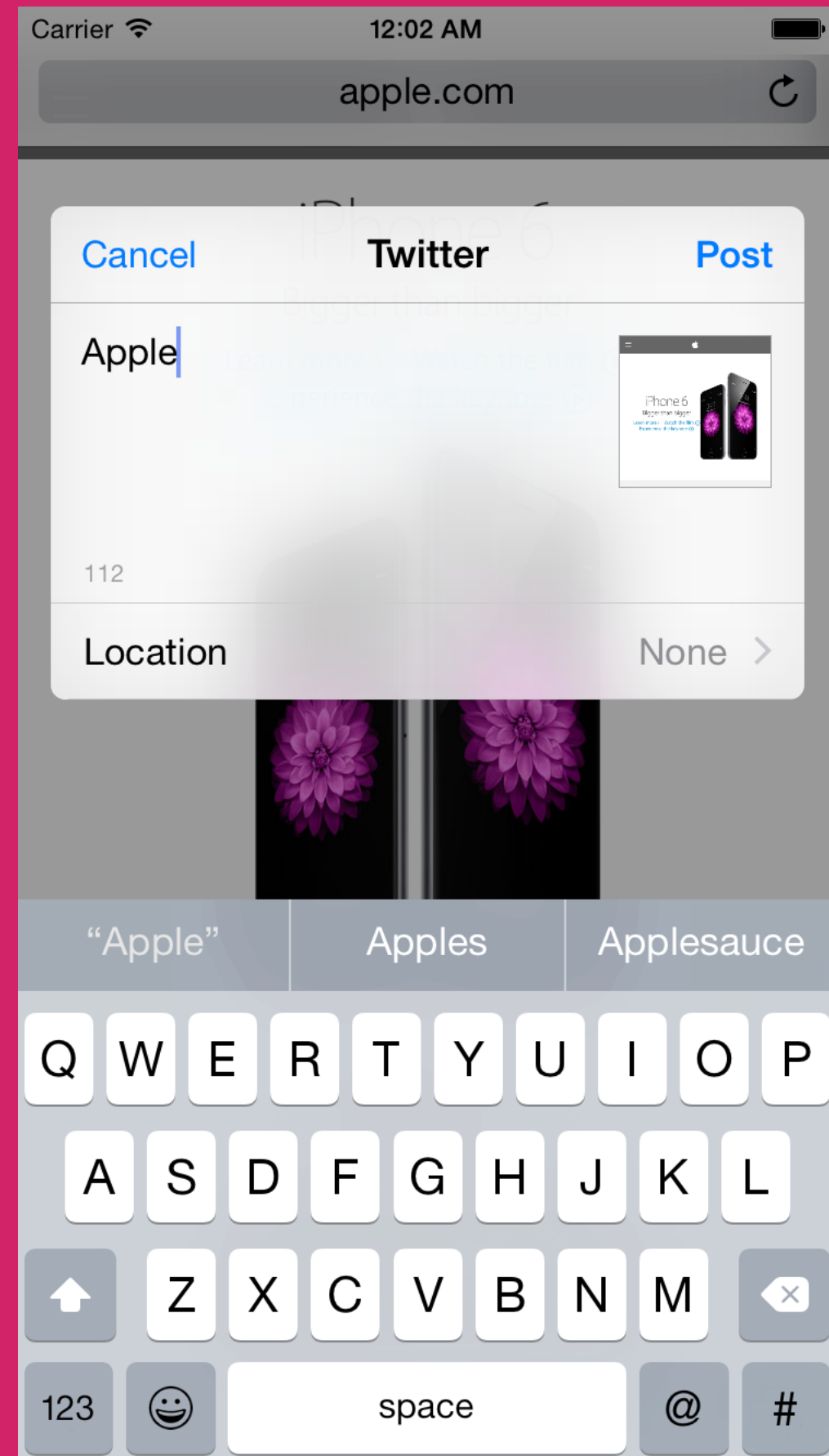


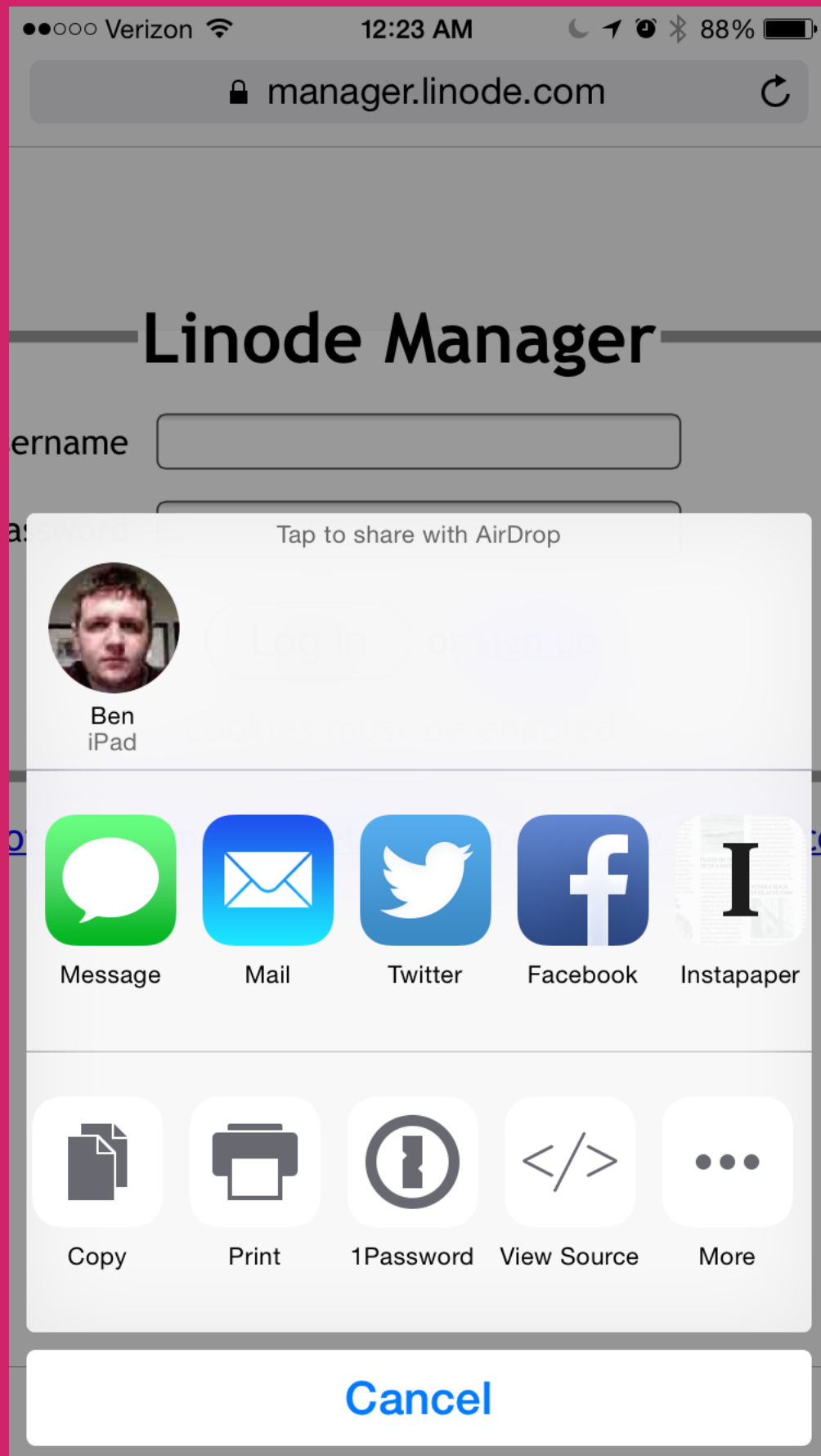
TODAY EXTENSION

Add Widget to Notification Center

SHARE EXTENSION

Share Content to Your App





ACTION EXTENSION

Act Upon Content

KEYBOARD

PHOTO EDITING

DOCUMENT PROVIDER

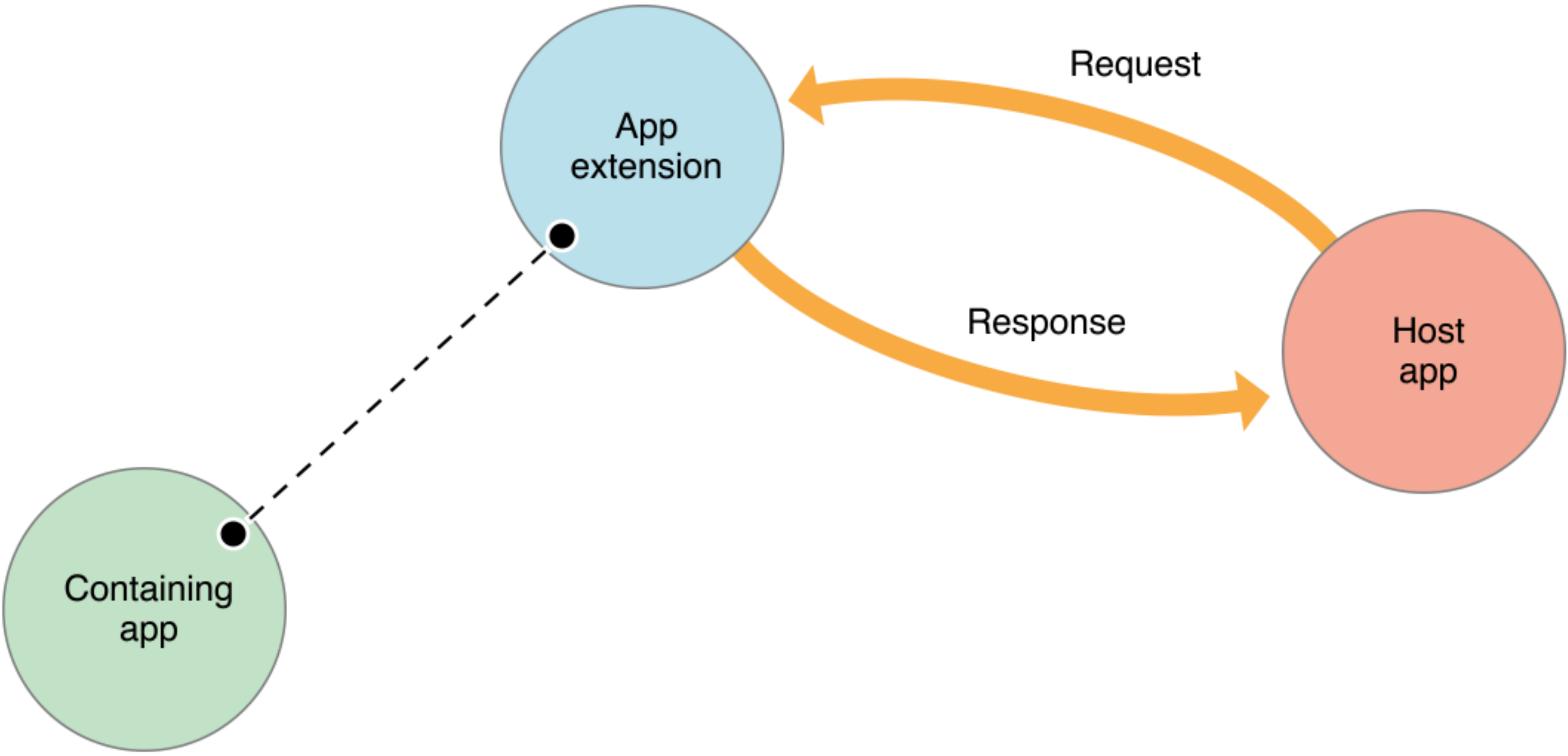
Not covered here :(

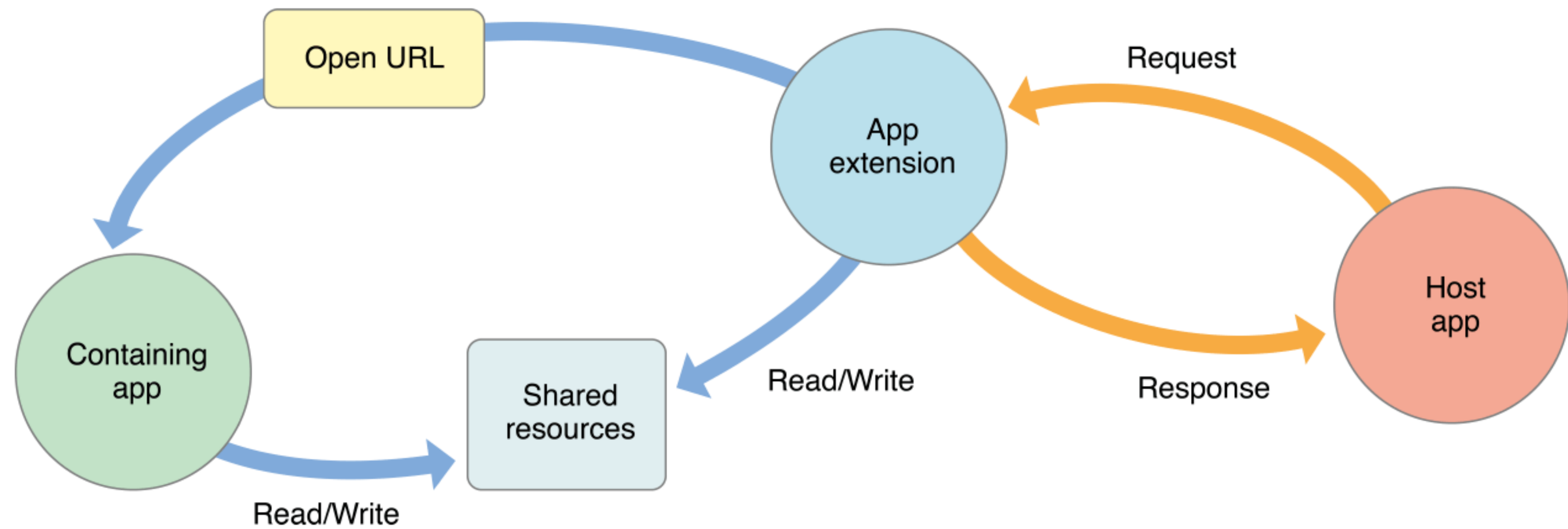
EXTENSION ARCHITECTURE

ISOLATED PROCESS

INDEPENDENT LIFECYCLE

DELIVERED WITH
APPLICATIONS





EXTENSIONS ARE JUST VIEW
CONTROLLERS

THE EVER-IMPORTANT INFO.PLIST

▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsWebURLWithMaxCount	String	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsWebURLWithMaxCount	String	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

NSExtension
top level dictionary

▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsWebURLWithMaxCount	String	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

NSExtensionAttributes

nested dictionary

▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsWebURLWithMaxCount	String	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

NSExtensionPrincipalClassName

Provide a class to instantiate if not using storyboards.

▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsWebURLWithMaxCount	String	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

NSExtensionMainStoryboard

Provide the name of your extension's storyboard.

▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsWebURLWithMaxCount	String	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

NSExtensionPointIdentifier

Indicates the type of extension to the OS.

WHAT IS AN ACTIVATION RULE?

SUPPORTED ACTIVATION RULES

NSExtensionActivationSupportsAttachmentsWithMaxCount
NSExtensionActivationSupportsAttachmentsWithMinCount
NSExtensionActivationSupportsFileWithMaxCount
NSExtensionActivationSupportsImageWithMaxCount
NSExtensionActivationSupportsMovieWithMaxCount
NSExtensionActivationSupportsText
NSExtensionActivationSupportsWebURLWithMaxCount
NSExtensionActivationSupportsWebPageWithMaxCount

OR A PREDICATE

Default is **TRUEPREDICATE**

DON'T SHIP WITH THIS!

```
SUBQUERY (  
    extensionItems,  
    $extensionItem,  
    SUBQUERY (  
        $extensionItem.attachments,  
        $attachment,  
        ANY $attachment.registeredTypeIdentifiers UTI-CONFORMS-TO "com.adobe.pdf"  
    ).@count == $extensionItem.attachments.@count  
).@count == 1
```

INFO.PLIST

Lots more options, read:

[https://developer.apple.com/library/prerelease/
iOS/documentation/General/Reference/
InfoPlistKeyReference/Articles/
SystemExtensionKeys.html](https://developer.apple.com/library/prerelease/iOS/documentation/General/Reference/InfoPlistKeyReference/Articles/SystemExtensionKeys.html)

SHARING CODE BETWEEN APP AND EXTENSIONS

Easy way:
just add a class to multiple targets!

app



extension



CoffeeTracker
CoffeeTracker
CoffeeTracker

Target Membership

- ☒  CoffeeTracker
- ☐  CoffeeTrackerTests
- ☒  CoffeeToday

Text Settings

Text Encoding Unicode (U

Line Endings Default – O

Though, you probably want to create a
Framework

Choose a template for your new target:

iOS - Application Framework & Library - Application Extension - Other - AFNetworking Temp... - NSScreencast Templ... OS X - Application - Framework & Library - System Plug-in - Other - AFNetworking Temp... - NSScreencast Templ...	 Cocoa Touch Framework  Cocoa Touch Static Library Cocoa Touch Framework This template creates a framework that uses UIKit.
--	---

Cancel

Deployment Target: 8.0

Devices: iPhone

Previous

Next

nt Info

Deployment Target

Devices

App Extensions ☒ Allow app extension API only

Frameworks and Libraries

Name

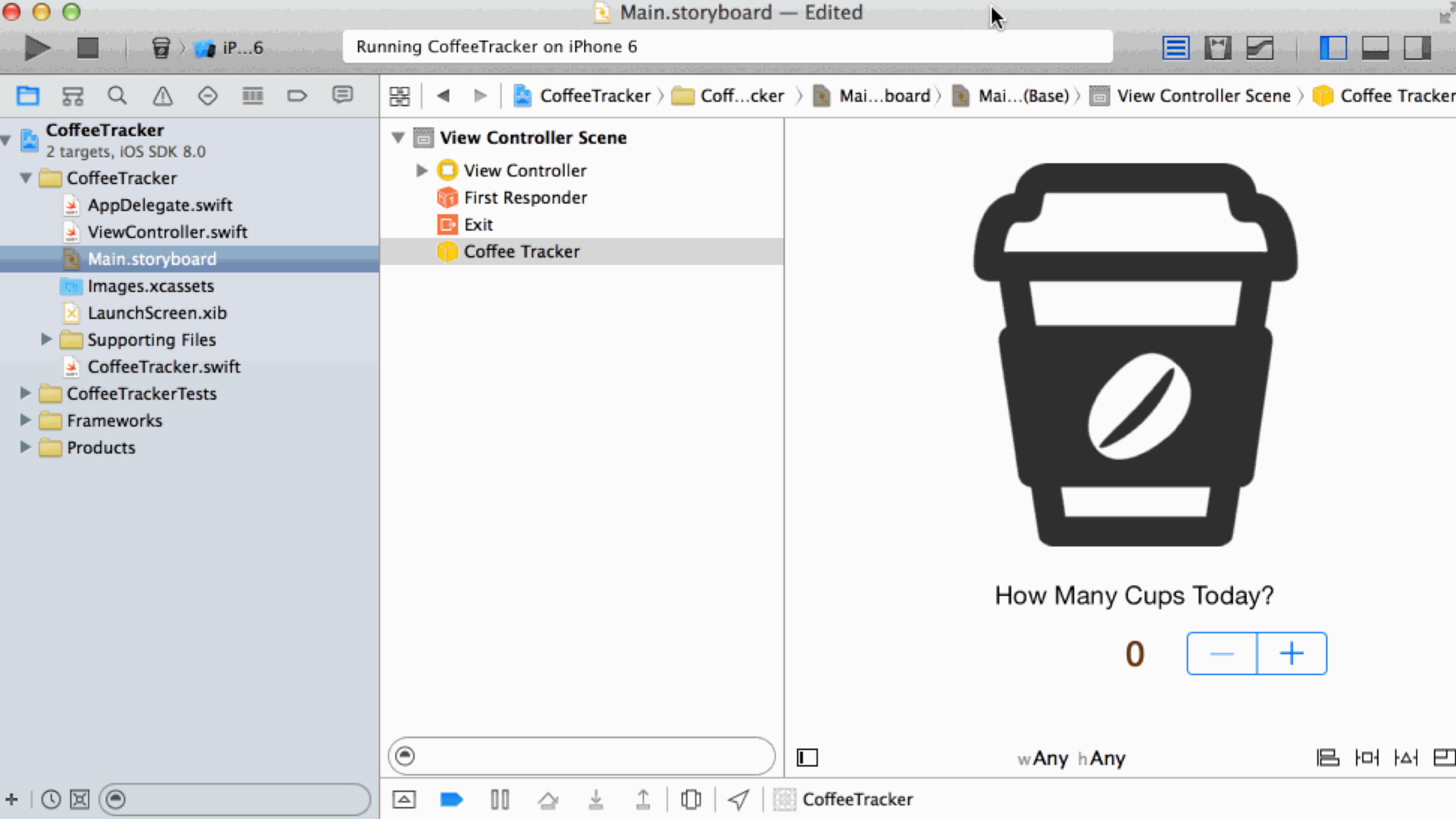
Sta

Add frameworks & libraries here

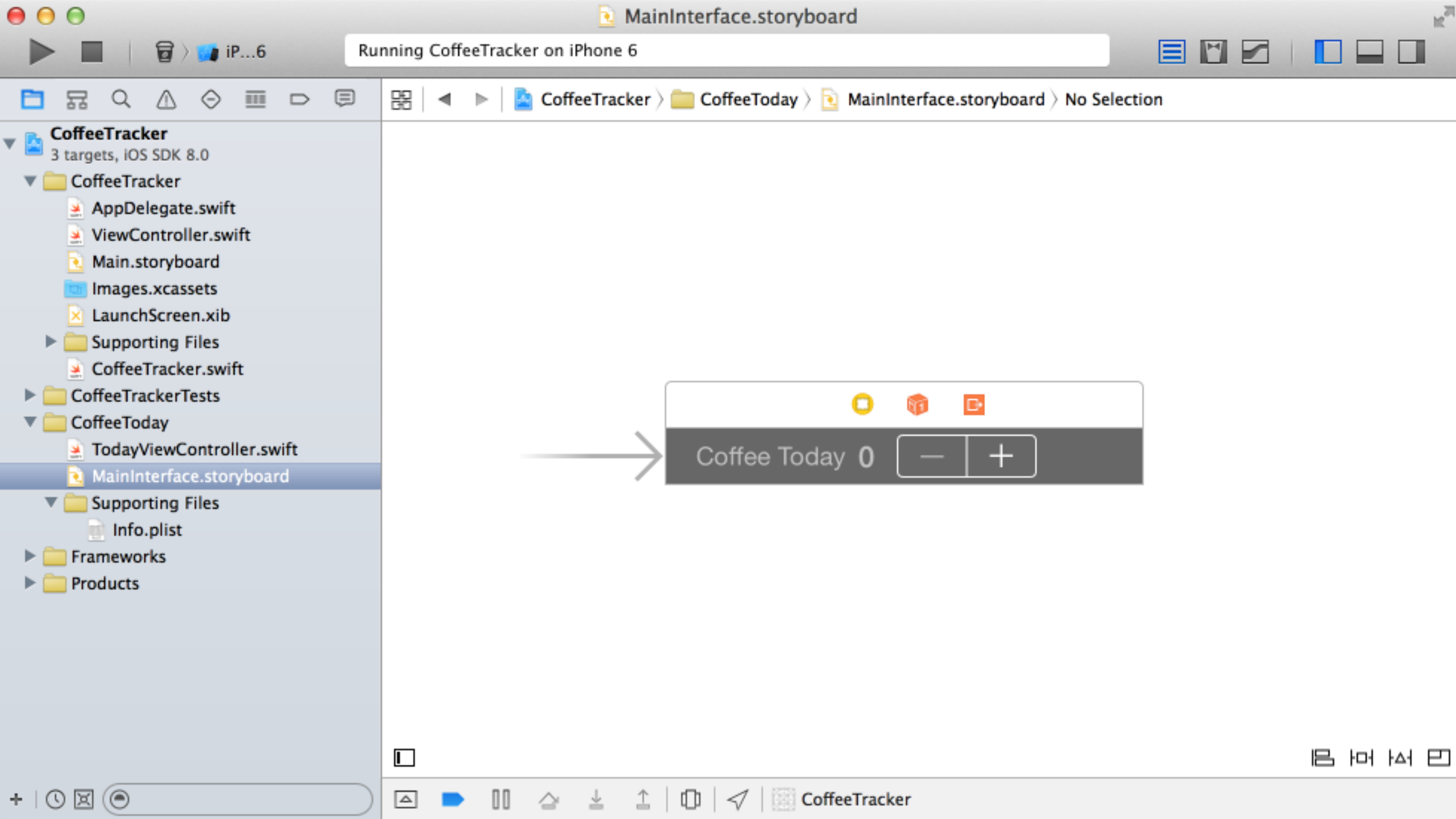
SHARING STORYBOARDS?

LET'S BUILD A TODAY EXTENSION

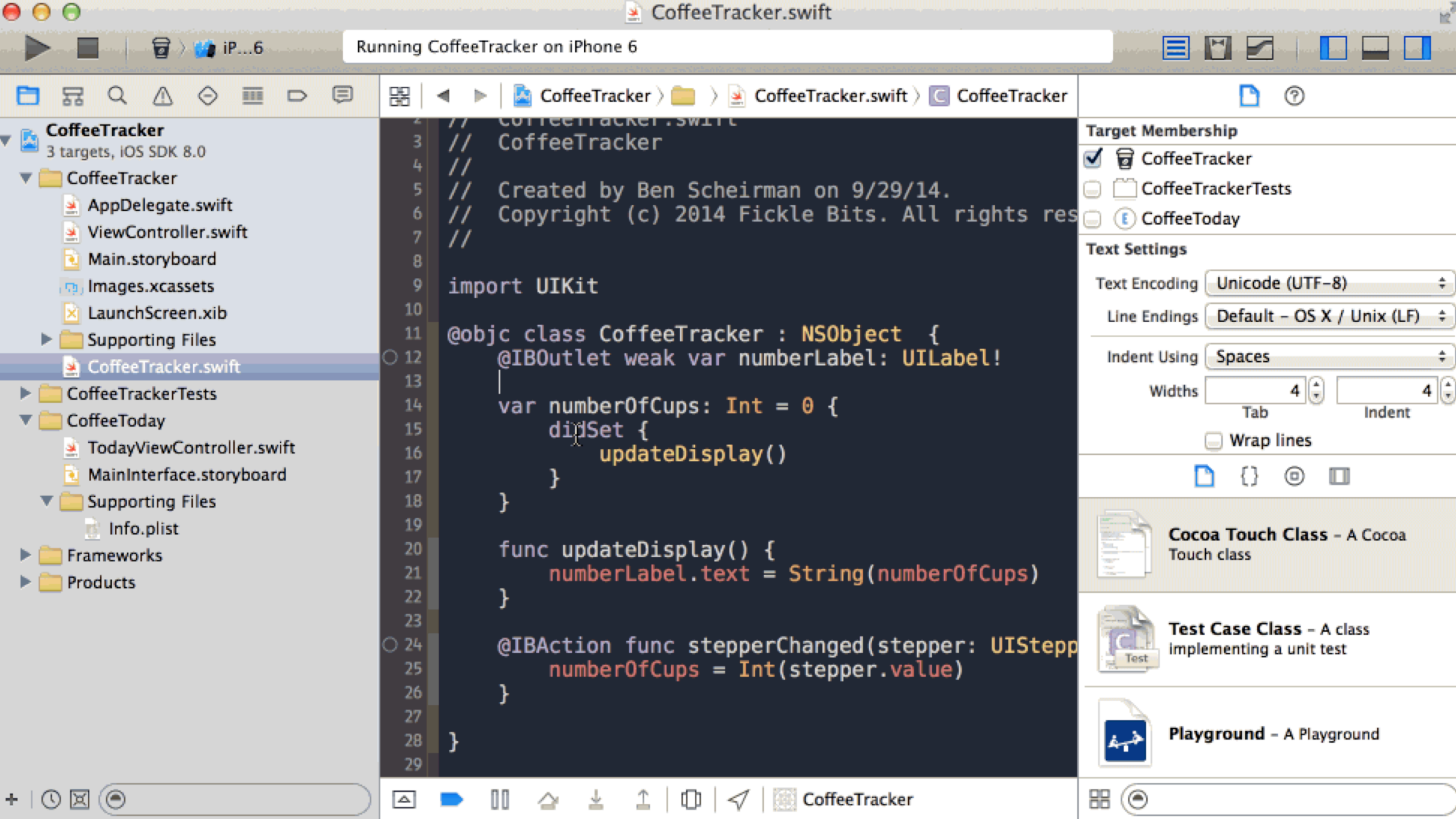
STEP 1: ADD EXTENSION
TARGET



STEP 2: DESIGN THE WIDGET



STEP 3: SHARE COMMON
CODE



STEP 4: WIRE IT UP

Today View Controller Scene

- Today View Controller
- First Responder
- Exit



No Selection



Object - Provides a template for objects and controllers not directly available in Interface Builder.



LET'S SEE
IT IN ACTION!



Calendar



Photos



Maps



Game Center



Newsstand



Health



Passbook



Settings

How do I keep them in sync?

SHARING DATA BETWEEN APP AND EXTENSION

CAN SHARE:

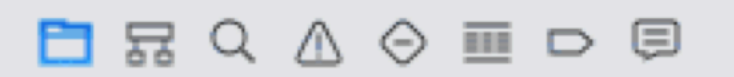
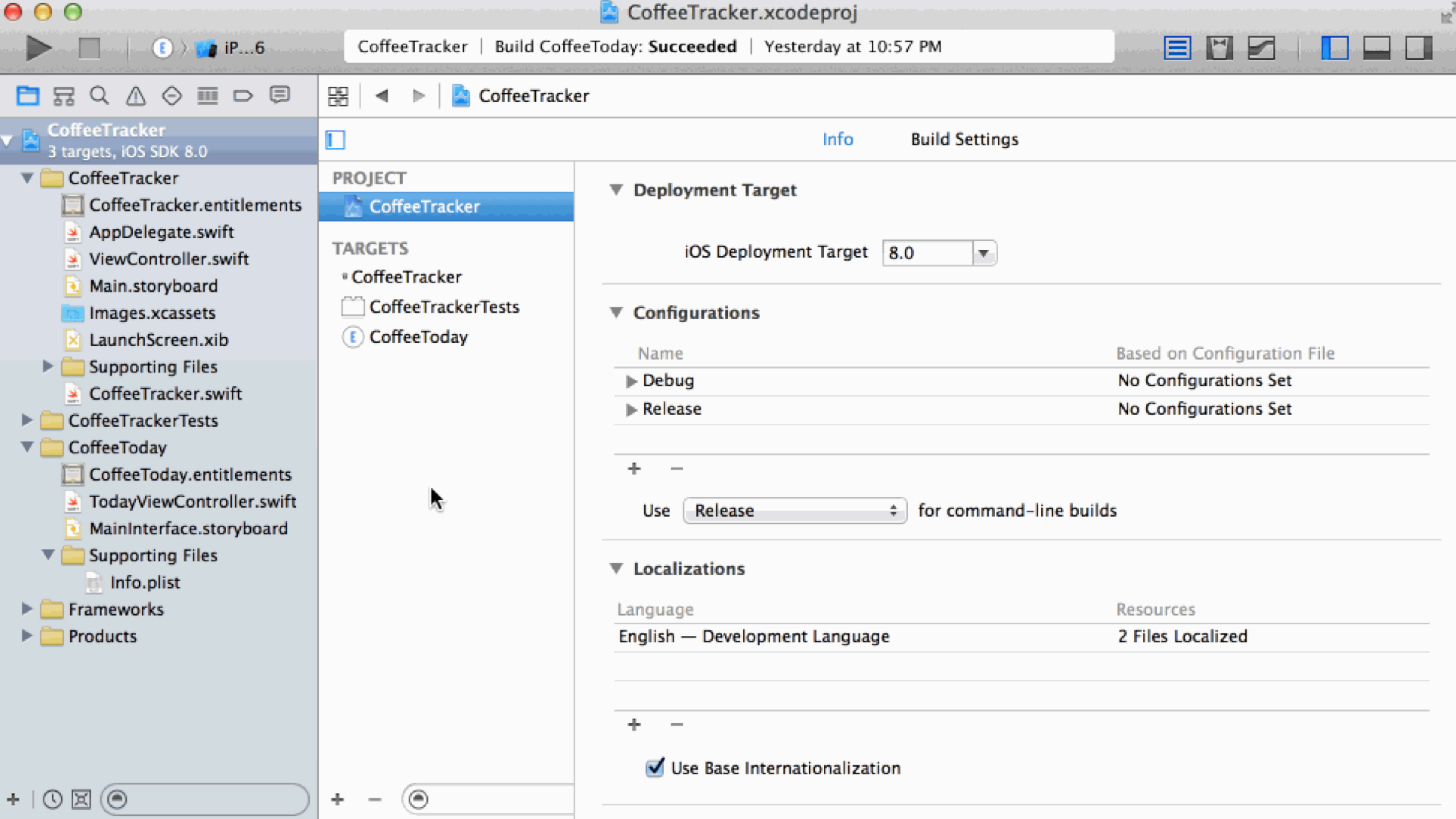
- **NSUserDefaults**
 - CoreData
 - Files
- **Background Transfer Sessions**

CREATE AN APP GROUP

must start with "group."

CREATE FOR CONTAINING APP

CREATE SAME GROUP FOR EXTENSION



▼ CoffeeTracker
3 targets, iOS SDK 8.0

- ▼ CoffeeTracker
 - CoffeeTracker.entitlements
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Images.xcassets
 - LaunchScreen.xib
 - Supporting Files
 - CoffeeTracker.swift
- ▼ CoffeeTrackerTests
- ▼ CoffeeToday
 - CoffeeToday.entitlements
 - TodayViewController.swift
 - MainInterface.storyboard
 - Supporting Files
 - Info.plist
- ▼ Frameworks
- ▼ Products



PROJECT

CoffeeTracker

TARGETS

- CoffeeTracker
- ☞ CoffeeTrackerTests
- ⓘ CoffeeToday

Info Build Settings

▼ Deployment Target

iOS Deployment Target 8.0 ▼

▼ 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	2 Files Localized

+ -
☒ Use Base Internationalization

SETUP SHARED NSUSERDEFAULTS

```
var userDefaults: UserDefaults {  
    get {  
        return UserDefaults(suiteName: "group.com.nsscreencast.coffeetracker")  
    }  
}
```

ADD TO COFFEE TRACKER

```
var numberOfCups: Int {  
    get {  
        return userDefaults.integerForKey("cups")  
    }  
    set {  
        userDefaults.setInteger(newValue, forKey: "cups")  
        updateDisplay()  
    }  
}
```

HANDLE PERIODIC BACKGROUND UPDATES

```
func widgetPerformUpdateWithCompletionHandler(completionHandler: (NCUpdateResult) -> ()) {  
    coffeeTracker.updateDisplay()  
  
    // If an error is encountered, use NCUpdateResult.Failed  
    // If there's no update required, use NCUpdateResult.NoData  
    // If there's an update, use NCUpdateResult.NewData  
  
    completionHandler(NCUpdateResult.NewData)  
}
```

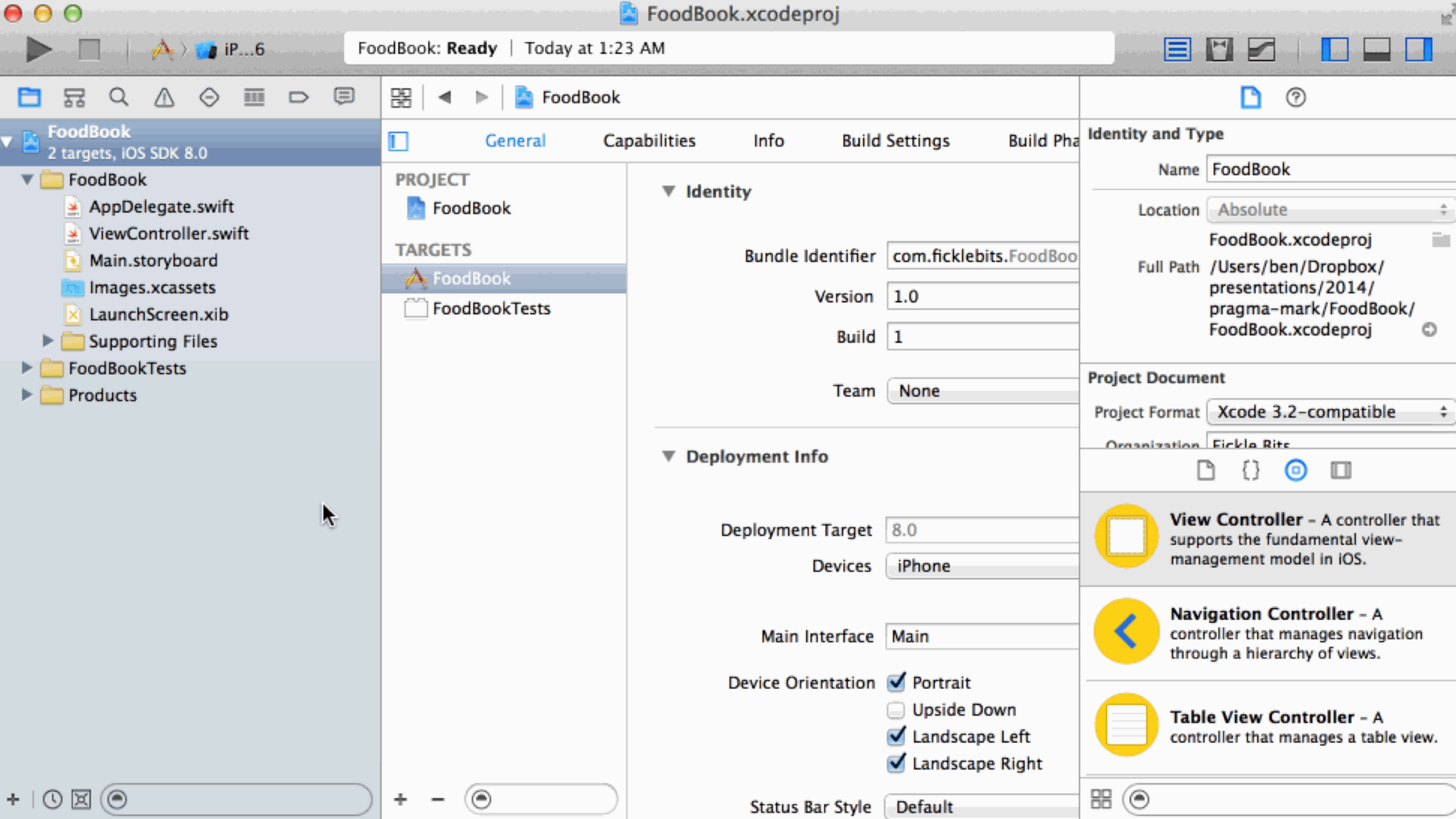
SHARE EXTENSIONS

- **Share content to another app / service**
 - **Use System Sharing UI or Custom**
- **Upload using shared background session**

INTRODUCING... **FOODBOOK**

STEP 1:

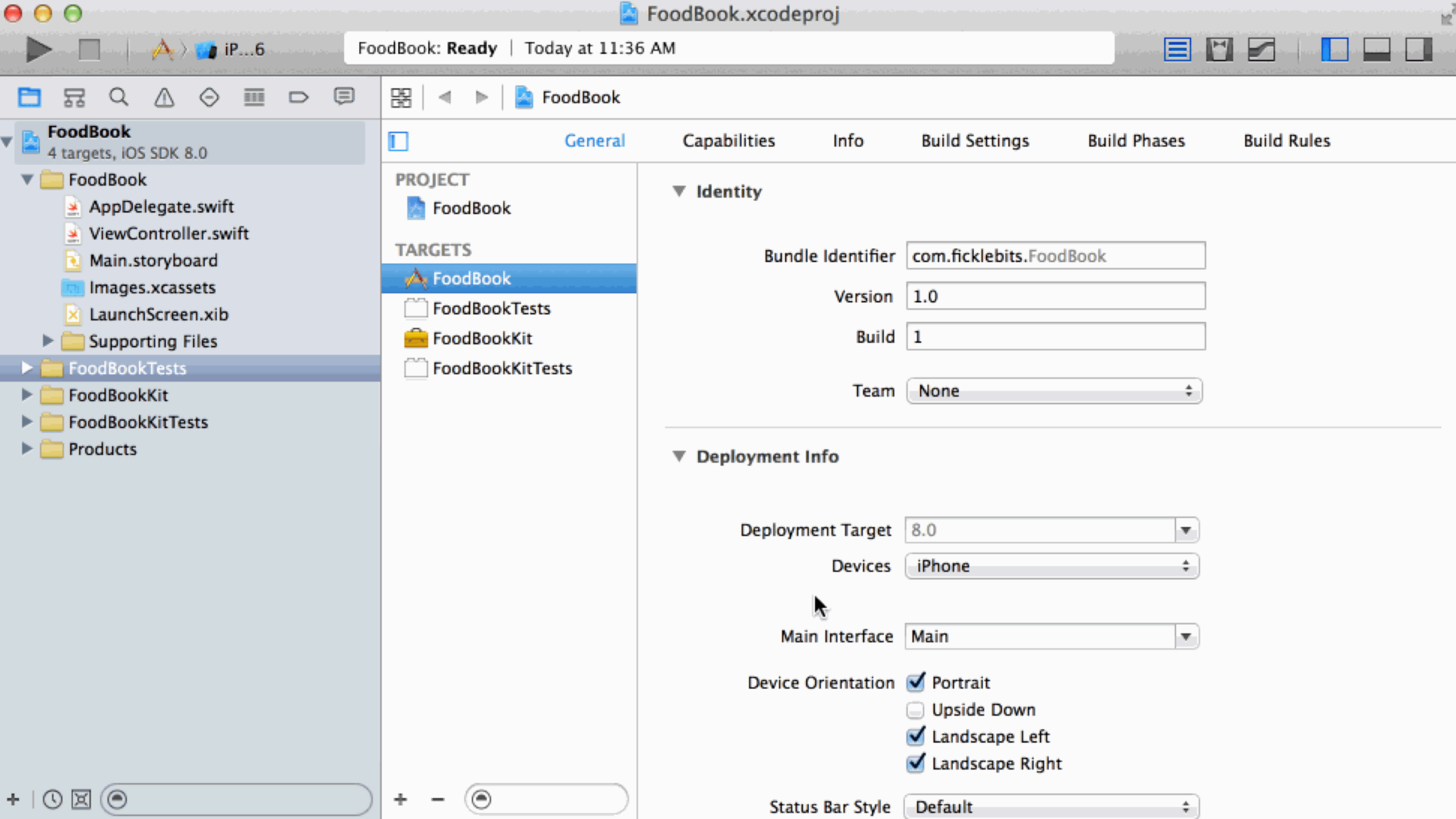
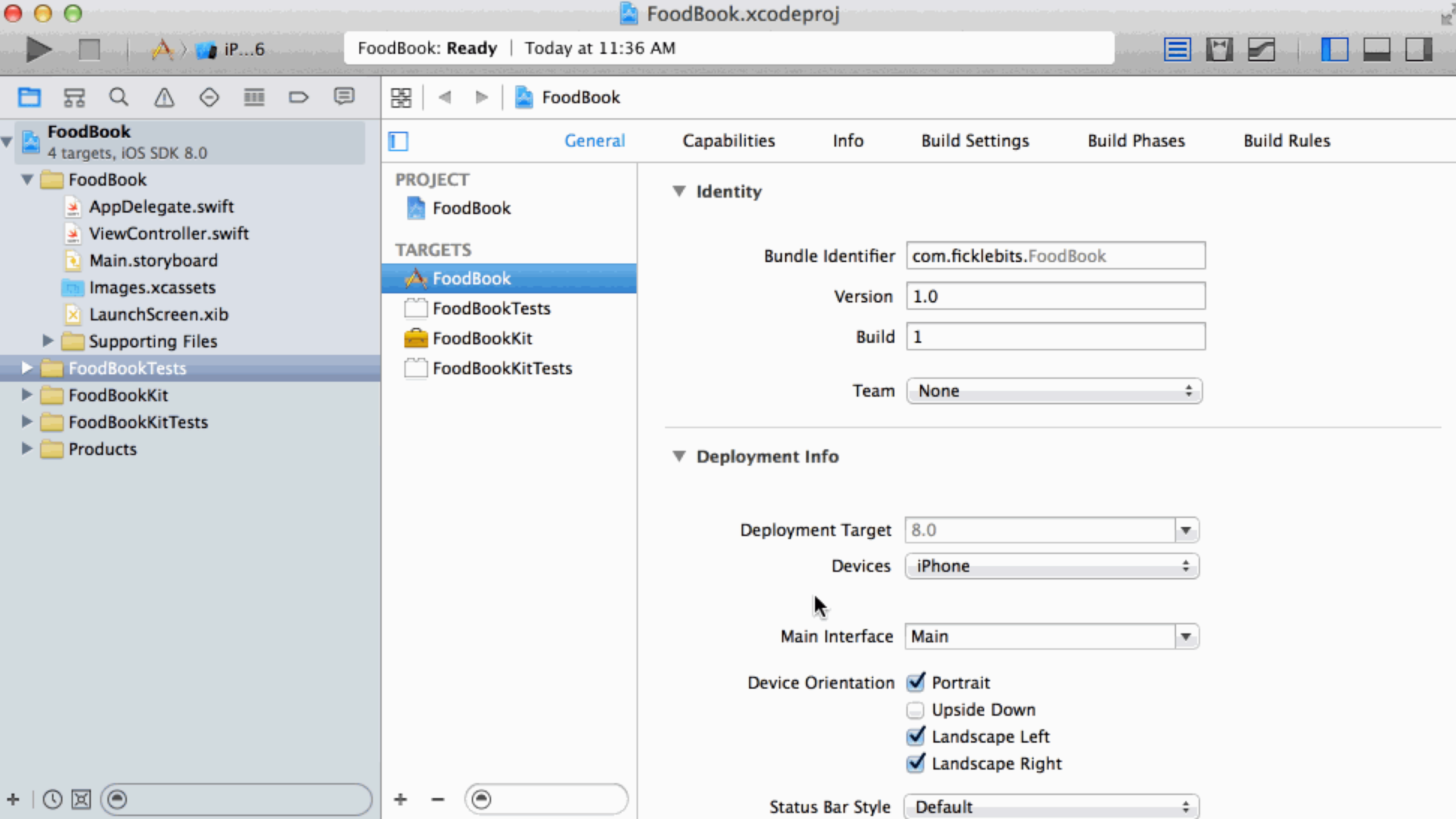
ADD A FRAMEWORK FOR OUR
SHARED CODE



**Useful for putting networking code or
view controllers common to app and
extension**

STEP 2:

ADD A SHARE EXTENSION
TARGET



STEP 3:

UPDATE INFO.PLIST

FoodBook

Build FoodBookShareExtension: **Succeeded**

Today at 11:53 AM

1

FoodBook > FoodBookShareExtension > Supporting Files > Info.plist > No Selection

Key	Type	Value
▼ Information Property List		
Localization native development region	String	en
Bundle display name	String	FoodBookShareExtension
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	com.ficklebits.FoodBook.\$(PRODUCT_NAME:rfc103)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
▼ NSExtension		
▼ NSExtensionAttributes	Dictionary	(1 item)
▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationSupportsImageWithMaxCount	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.share-services

FoodBook

5 targets, iOS SDK 8.0

FoodBook

AppDelegate.swift

ViewController.swift

Main.storyboard

Images.xcassets

LaunchScreen.xib

Supporting Files

FoodBookTests

FoodBookKit

FoodBookKitTests

FoodBookShareExtension

ShareViewController.swift

MainInterface.storyboard

Supporting Files

Info.plist

Products

STEP 4:

EDIT VIEW CONTROLLER

(BRACE YOURSELF)

IMPORT MOBILECORESERVICES

```
import MobileCoreServices
```

ADD A PROPERTY TO STORE THE IMAGE BEING SHARED

```
var image: UIImage?
```


GET THE ITEM PROVIDER

```
override fun viewDidLoad() {  
    var itemProvider: NSItemProvider?  
    if let items = extensionContext?.inputItems {  
        itemProvider = NSItemProvider(items)  
    }  
}
```


GET THE ITEM PROVIDER

```
override fun viewDidLoad() {  
    var itemProvider: NSItemProvider?  
    if let items = extensionContext?.inputItems {  
        if let item = items.first  
            as? NSExtensionItem {  
            // ...  
        }  
    }  
}
```

GET THE ITEM PROVIDER

```
override fun viewDidLoad() {  
    var itemProvider: NSItemProvider?  
    if let items = extensionContext?.inputItems {  
        if let item = items.first  
            as? NSExtensionItem {  
            if let attachment = item.attachments?.first  
                as? NSItemProvider {  
                itemProvider = attachment  
            }  
        }  
    }  
}
```


PULL AN IMAGE OUT OF THE ITEM PROVIDER

```
override fun viewDidLoad() {  
    // ...  
  
    let imageType = kUTTypeImage as NSString  
    if itemProvider?.hasItemConformingToTypeIdentifier(imageType) == true {  
        itemProvider?.loadItemForTypeIdentifier(imageType, options: nil) {  
            (item, error) in  
  
        })  
    }  
}
```

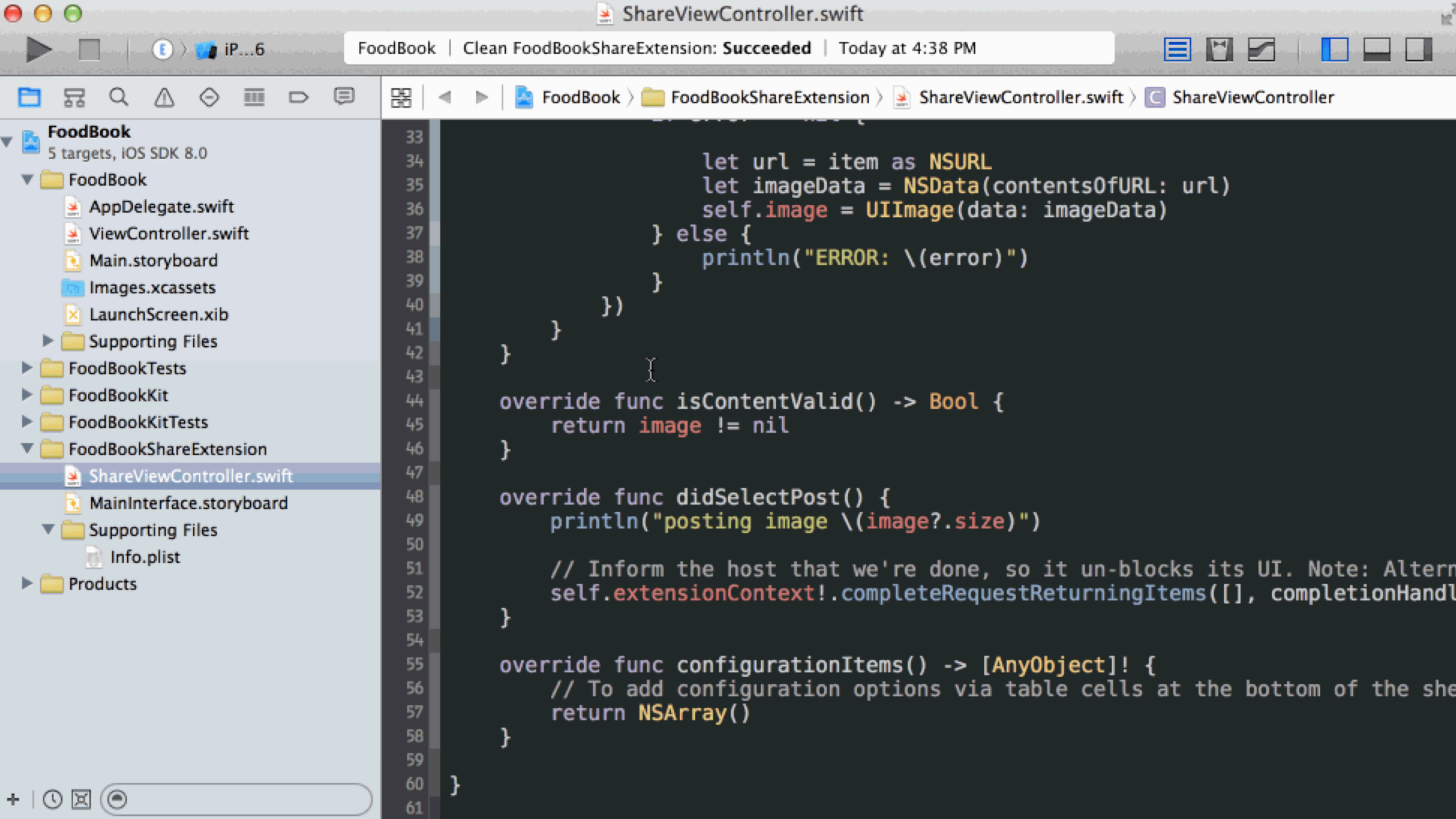
PULL AN IMAGE OUT OF THE ITEM PROVIDER

```
override func viewDidLoad() {  
    // ...  
  
    let imageType = kUTTypeImage as NSString  
    if itemProvider?.hasItemConformingToTypeIdentifier(imageType) == true {  
        itemProvider?.loadItemForTypeIdentifier(imageType, options: nil) {  
            (item, error) in  
            if error == nil {  
                let url = item as NSURL  
                let imageData = NSData(contentsOfURL: url)  
                self.image = UIImage(data: imageData)  
            } else {  
                println("ERROR: \(error)")  
            }  
        }  
    }  
}
```

STEP 5: PERFORM THE UPLOAD


```
override func didSelectPost() {  
    FoodBookApiClient.upload(image) {  
        self.extensionContext!.completeRequestReturningItems([],  
            completionHandler: nil)  
    }  
}
```

LET'S SEE
IT IN ACTION!



FoodBook | Clean FoodBookShareExtension: **Succeeded** | Today at 4:38 PM



FoodBook > FoodBookShareExtension > ShareViewController.swift > ShareViewController

- FoodBook
 - 5 targets, iOS SDK 8.0
 - FoodBook
 - AppDelegate.swift
 - ViewController.swift
 - Main.storyboard
 - Images.xcassets
 - LaunchScreen.xib
 - Supporting Files
 - FoodBookTests
 - FoodBookKit
 - FoodBookKitTests
 - FoodBookShareExtension
 - ShareViewController.swift
 - MainInterface.storyboard
 - Supporting Files
 - Info.plist
 - Products

```
33
34     let url = item as NSURL
35     let imageData = NSData(contentsOfURL: url)
36     self.image = UIImage(data: imageData)
37 } else {
38     println("ERROR: \(error)")
39 }
40 })
41 }
42 }
43 }
44 override func isValidContent() -> Bool {
45     return image != nil
46 }
47
48 override func didSelectPost() {
49     println("posting image \(image?.size)")
50
51     // Inform the host that we're done, so it un-blocks its UI. Note: Altern
52     self.extensionContext!.completeRequestReturningItems([], completionHandler
53 }
54
55 override func configurationItems() -> [AnyObject]! {
56     // To add configuration options via table cells at the bottom of the she
57     return NSArray()
58 }
59
60 }
61
```

CAN ALSO BRING YOUR OWN
VIEW CONTROLLER

ACTION EXTENSIONS

Act upon content user is looking at





Access to Entire DOM

Supply JavaScript that formats the **input** and
handles the **output**

Catify

2. *Language*, 1992, 29, 1, 1-12.

Key	Type	Value
-----	------	-------

Information Property List		Dictionary	(11 items)
Localization native development region	String	en	

Bundle display name	String	CatifyExtension
com.apple.catfish	com.apple.catfish	com.apple.catfish

Bundle identifier	String	com.ficklebits.Catify.\$(PRODUCT_NAME:rfc1
-------------------	--------	--

Bundle name	String	\$(PRODUCT_NAME)
-------------	--------	------------------

Bundle versions string, short	String	1.0
-------------------------------	--------	-----

Bundle version	String	1
----------------	--------	---

NSExtension	Dictionary (3 items)
NSExtensionAttributes	Dictionary (3 items)

▼ NSExtensionActivationRule	Dictionary	(1 item)
NSExtensionActivationRule	Dictionary	(1 item)

NSString	String	Action
NSExtensionJavaScriptPreprocessingFile	String	Action

NSExtensionPrincipalClass	String	\$(PRODUCT_MODULE_NAME).ActionRequest
---------------------------	--------	---------------------------------------

ACTION.JS

```
var Action = function() {};
```

```
Action.prototype = {
```

```
    run: function(arguments) {  
    },
```

```
    finalize: function(arguments) {  
    }
```

```
};
```

```
var ExtensionPreprocessingJS = new Action
```

```
run: function(arguments) {  
    arguments.completionFunction({  
        "host": window.location.host,  
        "path": window.location.pathname  
    })  
},
```

APP WILL RECEIVE
PROPERTYLIST ITEM TYPE

```
func beginRequestWithExtensionContext(context: NSExtensionContext) {
    self.extensionContext = context
    var itemProvider: NSItemProvider?

    if let item = context.inputItems.first as? NSExtensionItem {
        if let attachment = item.attachments?.first as? NSItemProvider {
            itemProvider = attachment
        }
    }

    let itemType = String(kUTTypePropertyList)
    if itemProvider?.hasItemConformingToTypeIdentifier(itemType) == true {
        itemProvider?.loadItemForTypeIdentifier(itemType, options: nil) {
            item, error in
            let dictionary = item as NSDictionary
            NSOperationQueue.mainQueue().addOperationWithBlock {
                let preprocessingResults = dictionary[NSExtensionJavaScriptPreprocessingResultsKey]
                    as [NSObject:AnyObject]
                self.itemLoadCompletedWithPreprocessingResults(preprocessingResults)
            }
        }
    } else {
        self.doneWithResults(nil)
    }
}
```

```
let dictionary = item as NSDictionary
NSOperationQueue.mainQueue().addOperationWithBlock {
    let results = dictionary[NSExtensionJavaScriptPreprocessingResultsKey]
    as NSDictionary
    self.itemLoadCompletedWithPreprocessingResults(results)
```

```
func itemLoadCompletedWithPreprocessingResults(results: NSDictionary) {  
    let host = results["host"] as String  
    let path = results["path"] as String  
    let newURL = "http://cat.\(host).meowbify.com\(path)"  
    doneWithResults(["newURL": newURL])  
}
```


PREPARE THE RESULTS TO GO BACK TO
THE JAVASCRIPT OBJECT

```
func doneWithResults(results: NSDictionary?) {  
    if let finalizeResults = results {  
        var resultsDictionary = [NSExtensionJavaScriptFinalizeArgumentKey: finalizeResults]  
        let itemType = kUTTypePropertyList as NSString  
        var resultsProvider = NSItemProvider(item: resultsDictionary, typeIdentifier: itemType)  
        var resultsItem = NSExtensionItem()  
        resultsItem.attachments = [resultsProvider]  
        extensionContext?.completeRequestReturningItems([resultsItem], completionHandler: nil)  
    } else {  
        extensionContext?.completeRequestReturningItems(nil, completionHandler: nil)  
    }  
  
    extensionContext = nil  
}
```


HANDLE THE RESULTS IN JAVASCRIPT

```
finalize: function(arguments) {  
    window.location = arguments["newURL"];  
}
```

OMG, CAN WE RUN IT YET?



Calendar



Photos



Maps



Game Center



Newsstand



Health



Passbook



Settings



Safari

QUESTIONS?

THANK YOU!

Slide links: <http://speakerdeck.com/subdigital/ios8-app-extension>

Code links: <https://github.com/subdigital/pragma-mark-2014>

Ben Scheirman

ben@scheirman.com

[@subdigital](#)

[@nsscreencast](#)