Kata Training - Practicing AppDevKit

Anistar Sung
Engineering Manager

Download Learning Template

git clone https://github.com/anistarsung/AppDevKitLearning.git

Agenda

- Leverage AppDevKit in few steps
- Build Some Apps with AppDevKit
- Using Swift with AppDevKit

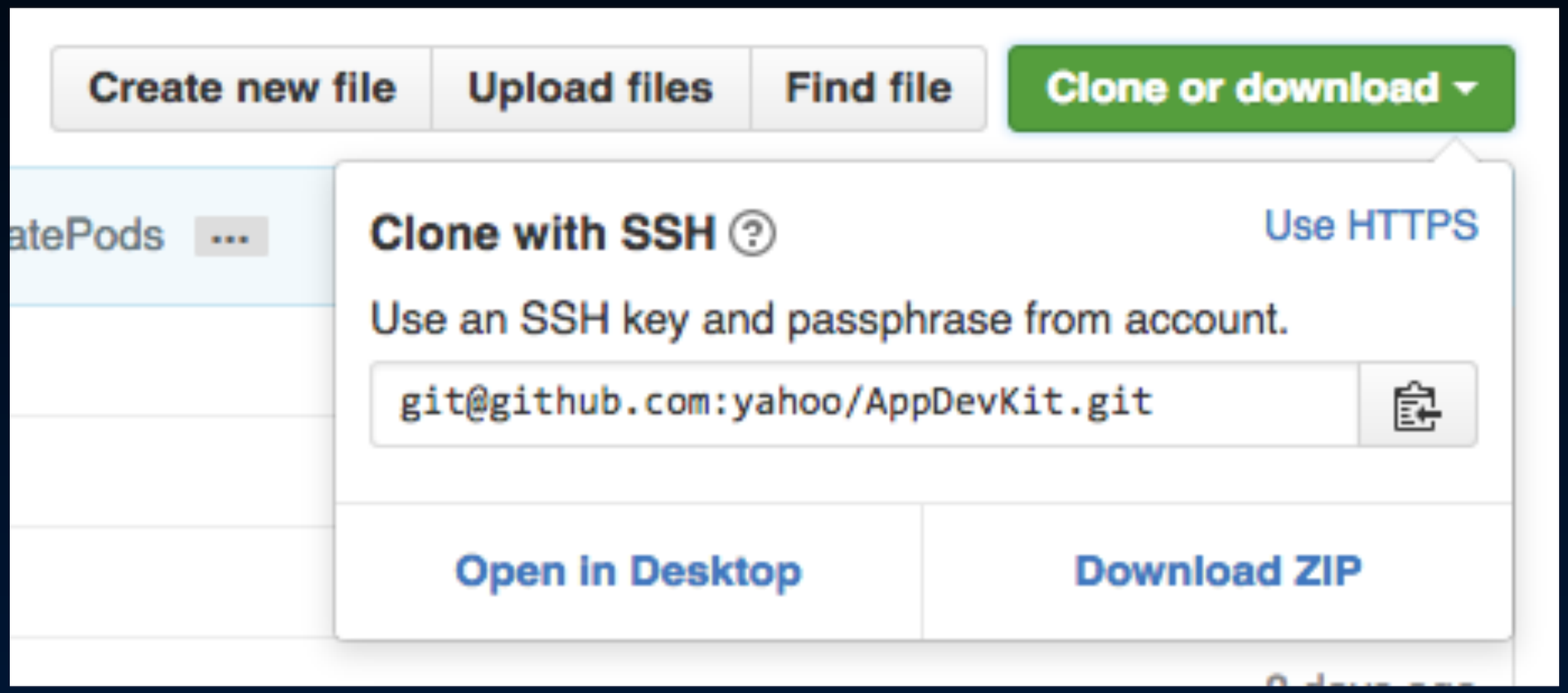
Leverage AppDevKit

Build First App with AppDevKit

Source Code of AppDevKit

https://github.com/yahoo/AppDevKit

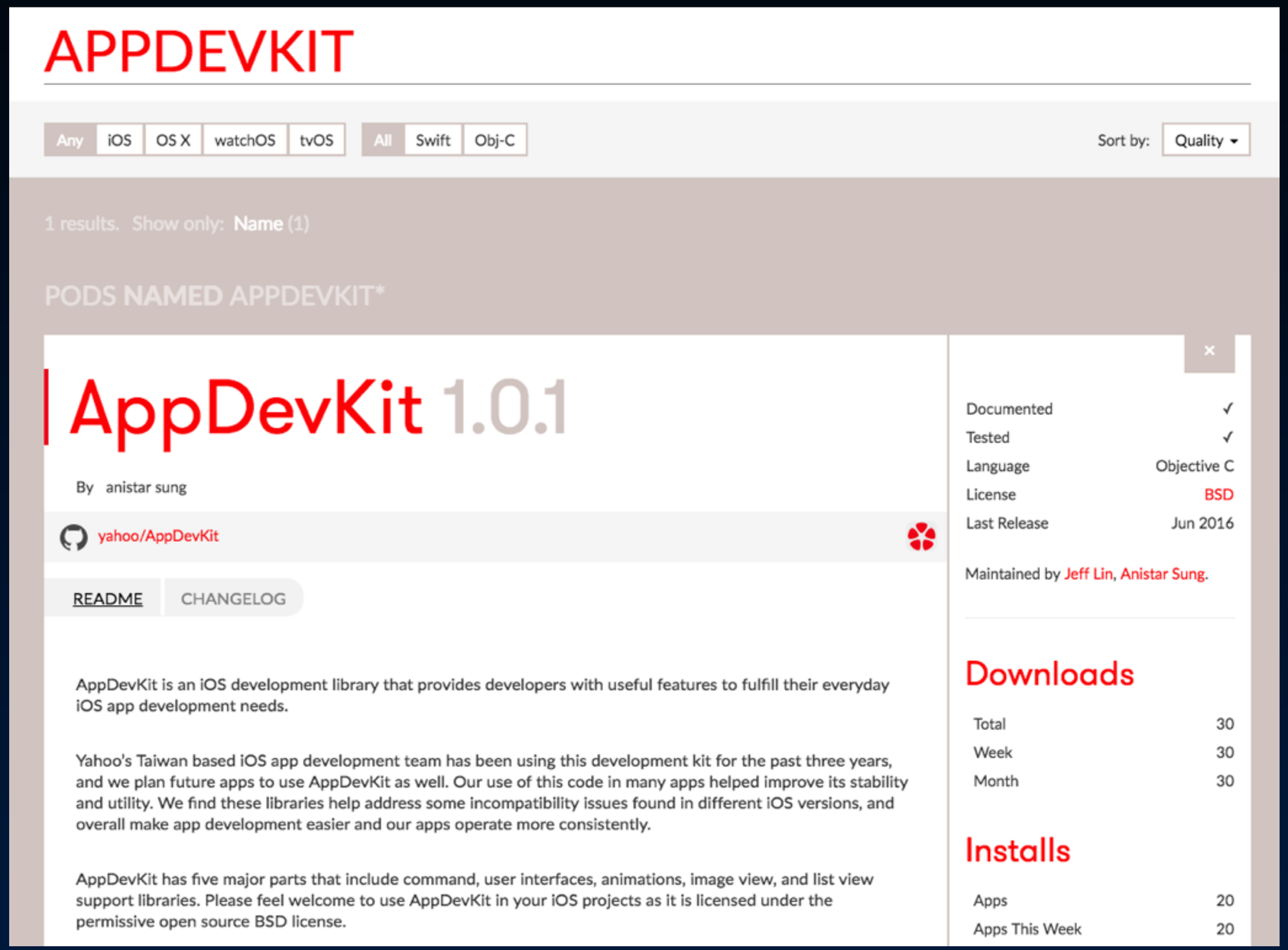
Get Source Code from Github



Set up AppDevKit

- Cloning code into your project directly
- Using CocoaPods to manage package

What’s CocoaPods



Using CocoaPods

Installing CocoaPods

```
$ sudo gem install cocoapods
```

Initializing CocoaPods

```
$ pod init
$ nano Podfile
```

Using CocoaPods

Editing Podfile

```
platform :ios, '8.0'

target 'MyApp' do
  pod 'AppDevKit', '~> 1.0’
  pod 'AFNetworking', '~> 2.6'
end
```

Using CocoaPods

Installing Package by CocoaPods

```
$ pod install
```

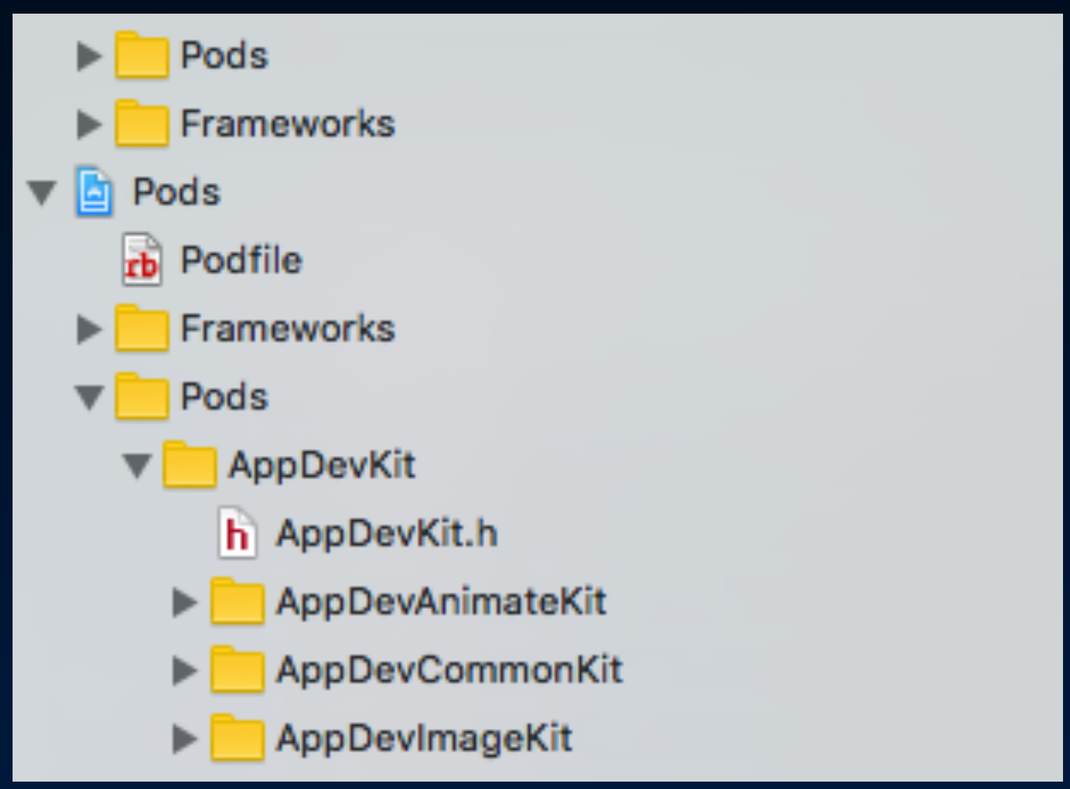
Updating Package by CocoaPods

```
$ pod update
```

Working with AppDevKit

Open iOS Project with Pods

```
$ open MyApp.xcworkspace
```



Working with AppDevKit

Importing Libraries from AppDevKit

```
#import <AppDevKit.h>
```

Demo

Start HelloWorld with AppDevKit

Using UIColor Hex presentation

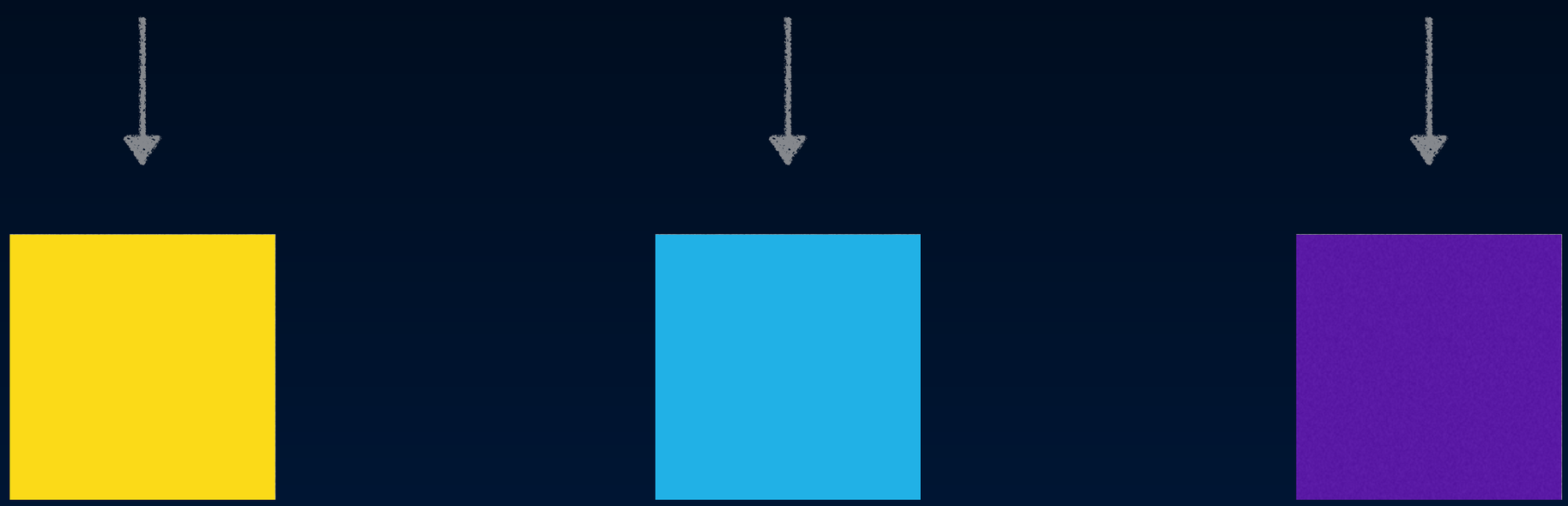
Utilities made setting color easier

Hex Color Format from Designer

? FBDA18

? 21B1E6

? 4C00A5



But UIColor….

```
+(UIColor *)colorWithWhite:(CGFloat)white alpha:(CGFloat)alpha;

+(UIColor *)colorWithHue:(CGFloat)hue saturation:
(CGFloat)saturation brightness:(CGFloat)brightness alpha:
(CGFloat)alpha;

+(UIColor *)colorWithRed:(CGFloat)red green:(CGFloat)green blue:
(CGFloat)blue alpha:(CGFloat)alpha;

......
```

No hex presentation… :’(

AppDevKit Hex Color Utility

```
[UIColor ADKColorWithHexString:@"fbda18"]
```

```
[[UIColor ADKColorWithHexString:@"fbda18"]
      ADKColorShiftByBrightness:0.7f];
```

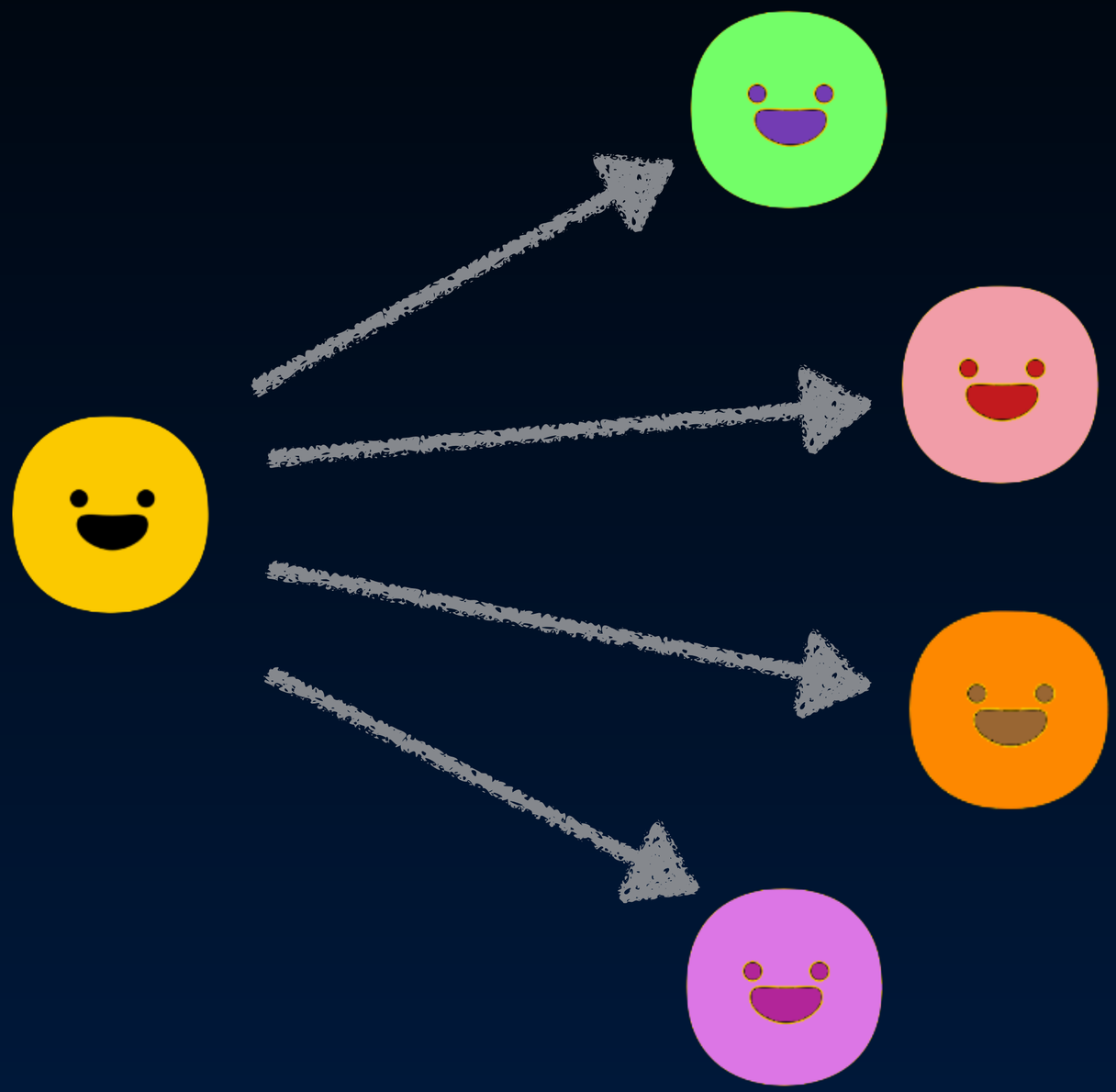
Demo

Using UIColor+ADKHexPresentation

Applying UIImage Filters

Changing image color in one line

Color Replacement



Color Replacement

- One image could fulfill all states change
- Reducing App size
- Providing other filters and color replacement methods

Color Replacement

```
[UIImage ADKImage:originalImage
replaceColor:[UIColor redColor]];
```

Demo

Using UIImage+ADKImageFilter

Using Animation Marcos

Animation is a piece of cake

Animation Marcos

Animation Marcos

```
[button popUpAnimation];
```

Demo

Using UIView+ADKAnimationMacro

Using Modal Mask View

Making A Focused UX Flow

Modal Mask View

Modal Mask View

- Presenting custom modal view and dismiss automatically
- Below iOS 8, it don’t support UIModalPresentationOverCurrentContext

Modal Mask View

```
[[ADKModalMaskView alloc] initWithView:UIView
modalColor:UIColor autoDismiss:YES];

[modalView showInView:UIView withAnimation:YES
completion:^(BOOL finished){…}];
```

Demo

Using ADKModalMaskView

Drawing Gradient Background

Without Bitmap Image Solution

ADKGradientView

ADKGradientView

- Don’t need bitmap image (Programmable solution)
- Reduce App size
- Support assigning color and orientation

ADKGradientView

```
view.beginColor = [UIColor redColor];
view.endColor = [UIColor greenColor];
view.blendsType = ADKBlendsTypeFromLeftTopToRightBottom;
```

ADKBlendsTypeFromTopToBottom
ADKBlendsTypeFromLeftToRight
ADKBlendsTypeFromLeftTopToRightBottom
ADKBlendsTypeFromRightTopToLeftBottom

Demo

Using ADKGradientView

Using AutoLayout Utility

Controlling AutoLayout easier

AutoLayout Utility

AutoLayout Utility

- Resolving elements aligning problem

Original

Unexpected

Expected

AutoLayout Utility

```
[view ADKHideViewWidth];

[view ADKHideTrailingConstraint];
```

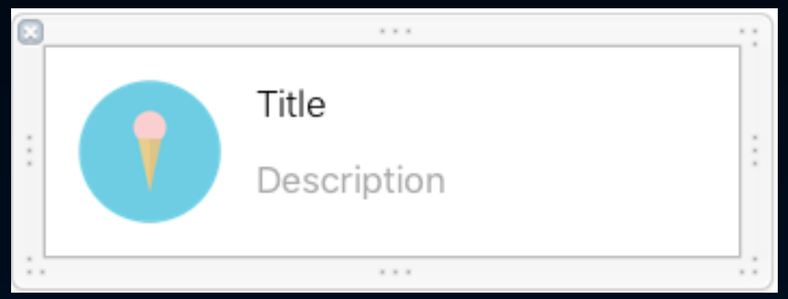
Demo

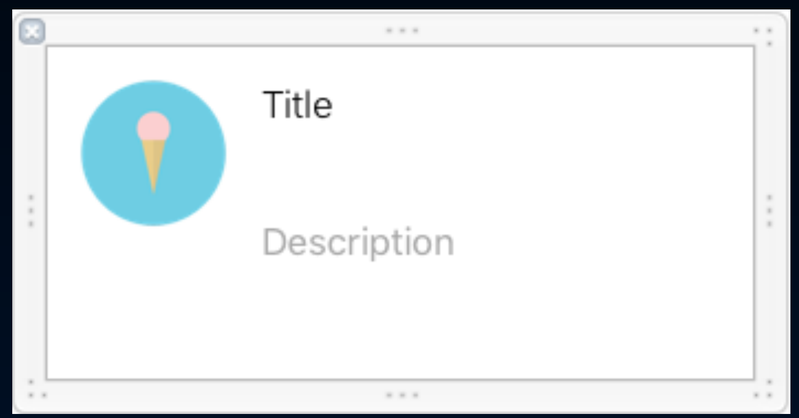
Using UIView+AutoLayoutSupport

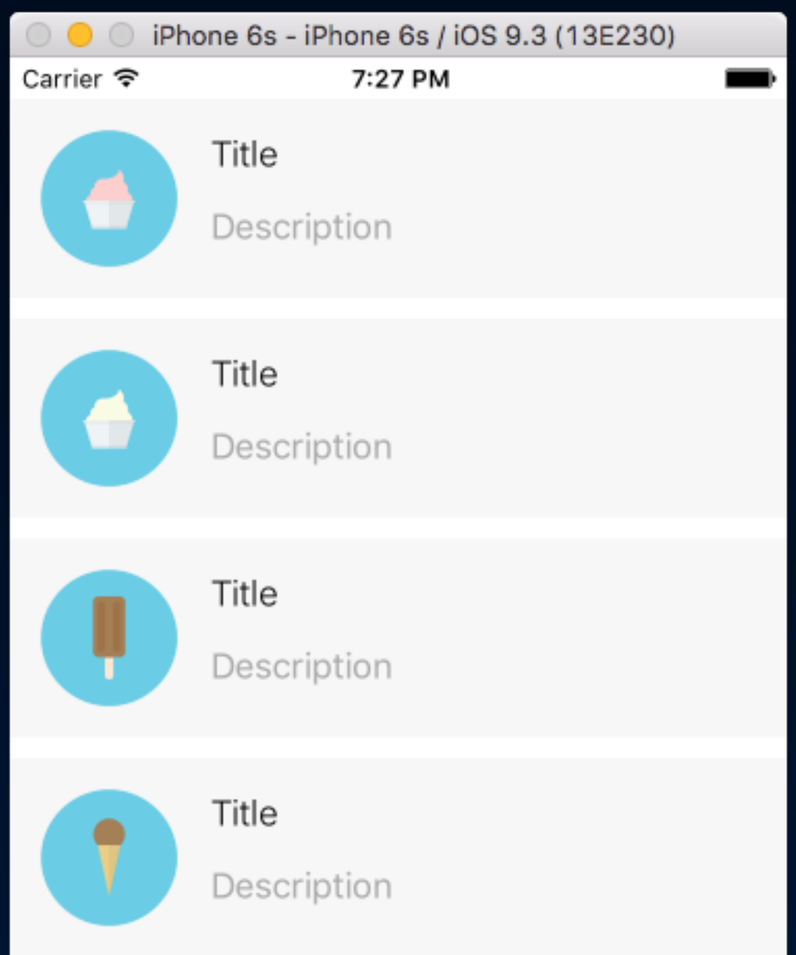
Using ADKNibSizeCalculator

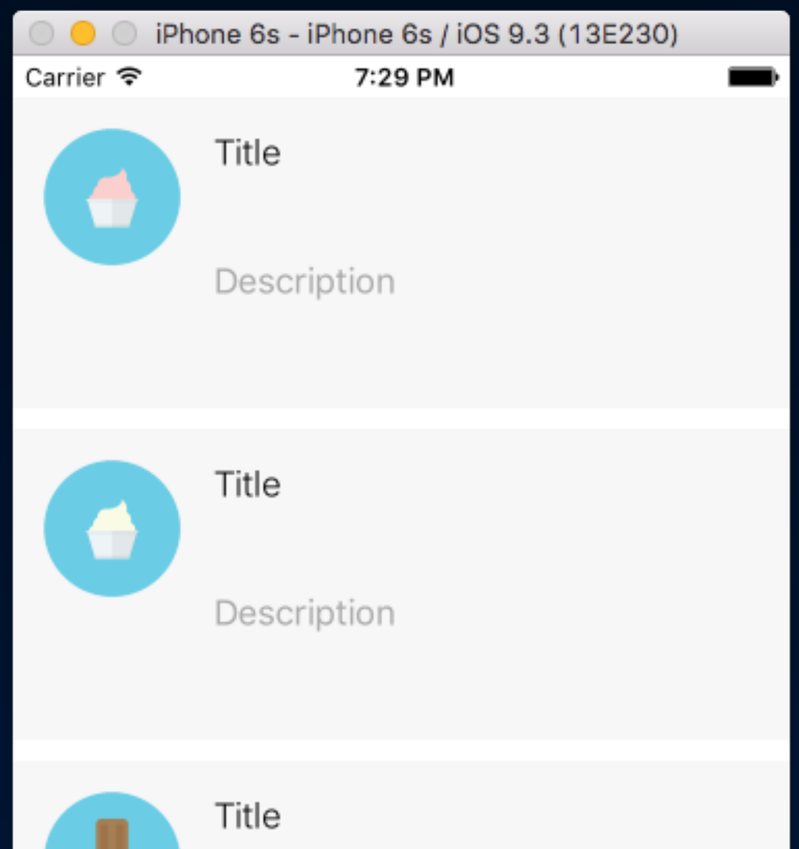
Nice tool for Communication

ADKNibSizeCalculator



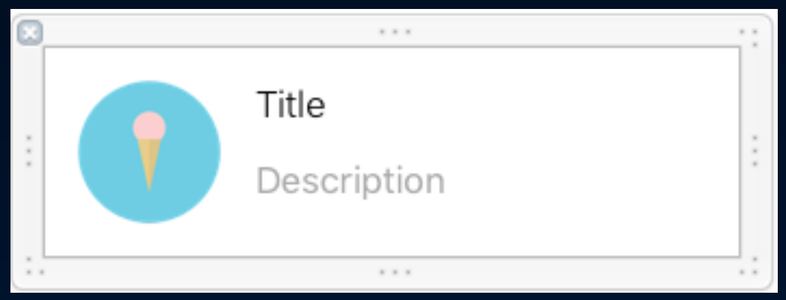






ADKNibSizeCalculator

Modify nib file without changing any code



Using ADKNibSizeCalculator

Use ADKNibSizeCalculator to calculate size

```
CGSize size = [[ADKNibSizeCalculator sharedInstance]
sizeForNibNamed:@“NibName” withStyle:ADKNibFixedHeightScaling];
```

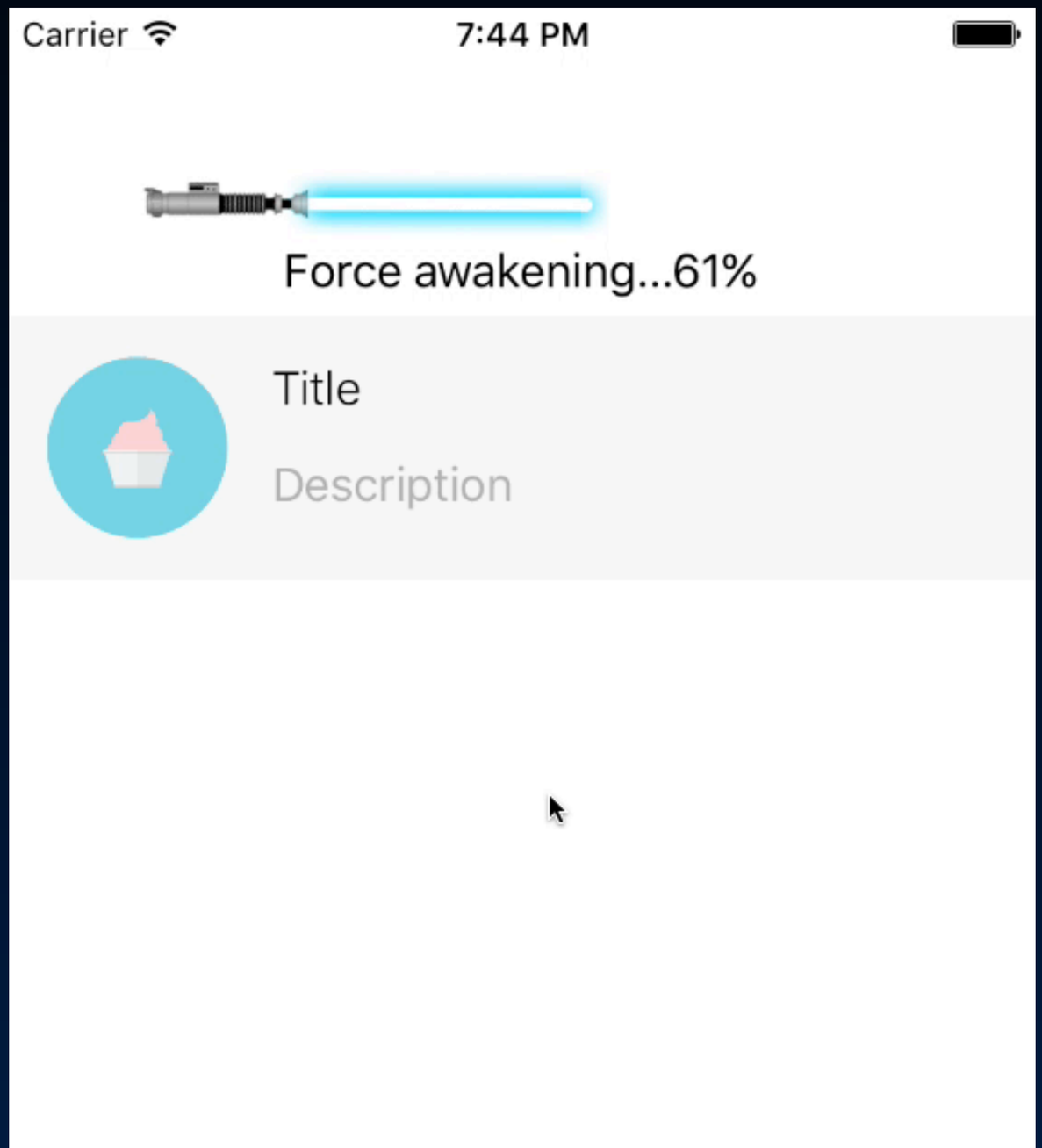
Demo

Calculating Nib Size with AppDevKit

Pull To Refresh in List View

Solution for Pull To Refresh

UIScrollView+ADKPullToRefreshView



Implement a PullToRefresh View

Create a view which implemented protocol ADKPullToRefreshViewProtocol

```
#import <AppDevKit/UIScrollView+ADKPullToRefreshView.h>

@interface PullToRefreshView : UIView <ADKPullToRefreshViewProtocol>
```

Implement required method for this protocol

```
- (void) ADKPullToRefreshStopped:(UIScrollView *)scrollView
- (void) ADKPullToRefreshTriggered:(UIScrollView *)scrollView
- (void) ADKPullToRefreshLoading:(UIScrollView *)scrollView
```

And more optional method for this protocol

```
- (CGFloat) ADKPullToRefreshTriggerDistanceTimes:(UIScrollView *)scrollView
- (void) ADKPullToRefreshDragging:(UIScrollView *)scrollView
- (void) ADKPullToRefreshView:(UIScrollView *)scrollView
  draggingWithProgress:(CGFloat)progress
```

Add PullToRefresh View on Scroll View

Add PullToRefresh View on Scroll View

```
[scrollView ADKAddPullToRefreshWithHandleView:refreshView
actionHandler:^{
   // do something here
}];
```

Don’t forget to stop animation after things done

```
[scrollView.pullToRefreshContentView stopAnimating];
```

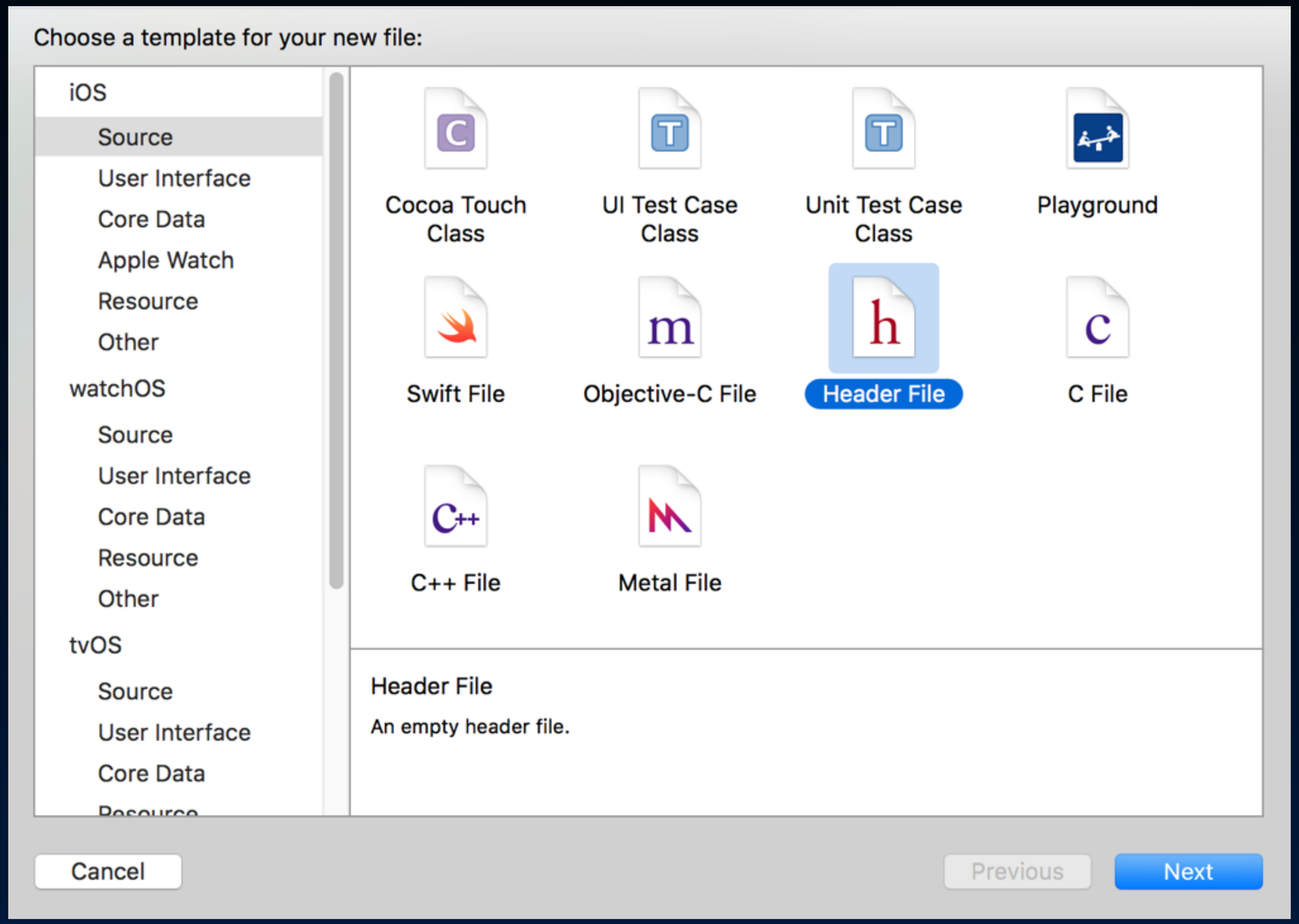
Demo

Implement Pull To Refresh with AppDevKit

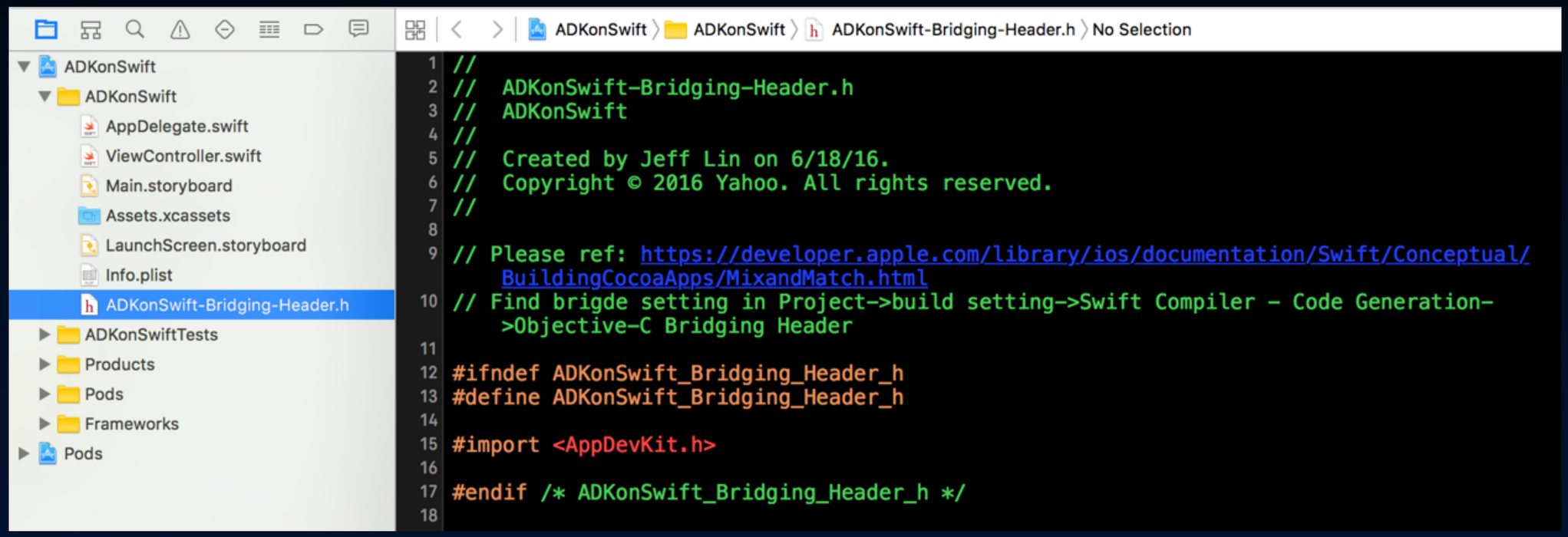
Using AppDevKit in Swift project

AppDevKit Works Well with Swift

Add Bridge Header

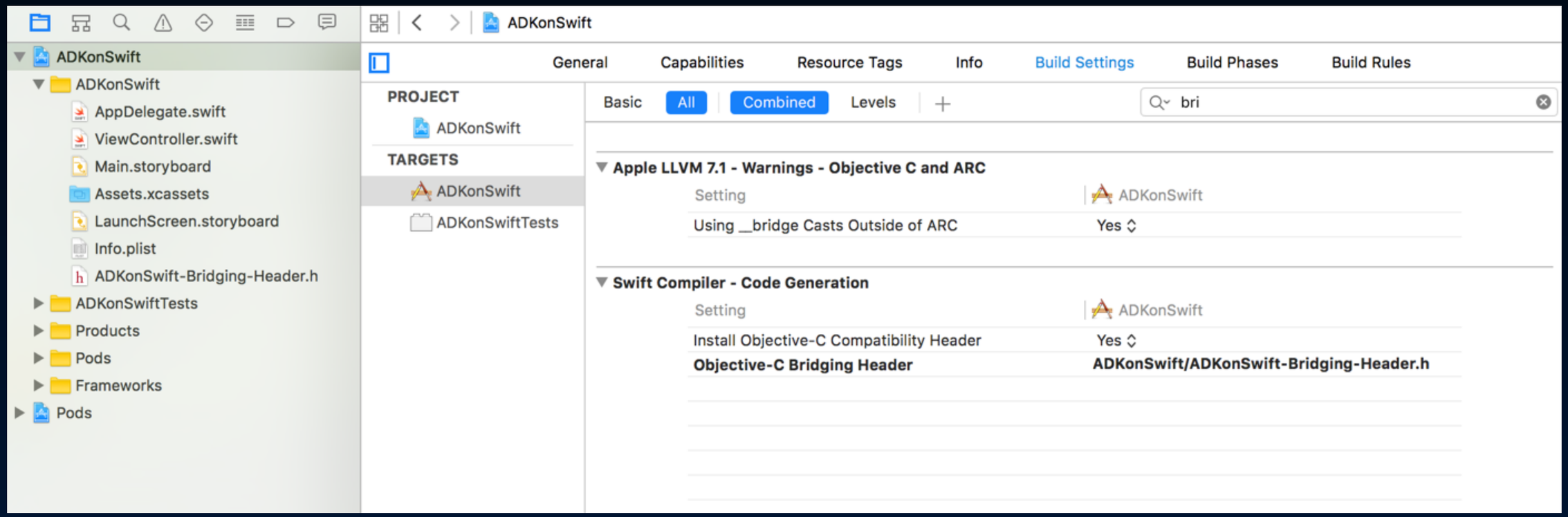


Add Bridge Header




```
#import <AppDevKit.h>
```

Bridge Header Settings



Using Swift with AppDevKit

```swift
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        self.setupView()
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    func setupView() -> Void {
        self.view.backgroundColor = UIColor.ADKColorWithHexString("0x1DB2E8")
    }
}
```

```swift
self.view.backgroundColor =
UIColor.ADKColorWithHexString("0x1DB2E8")
```

Q + A