

#WWDC19

# Great Developer Habits

Josh Tidsbury, Technology Evangelism



craft

\ 'kraft \

1. skill in planning, making, or executing
2. to make or produce with care, skill, or ingenuity







Hidden details matter.

Organize

Track

Document

Test

Analyze

Evaluate

Decouple

Manage



Organize

Track

Document

Test

Analyze

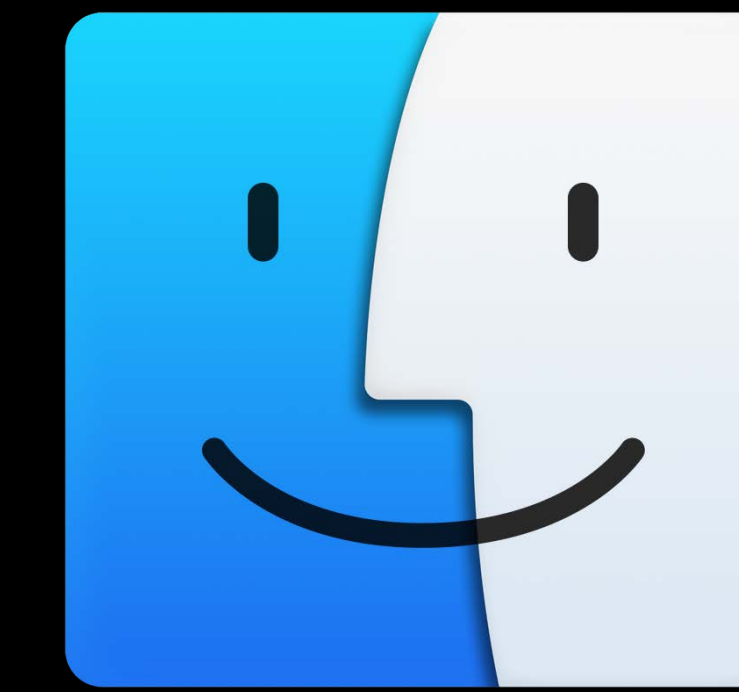
Evaluate

Decouple

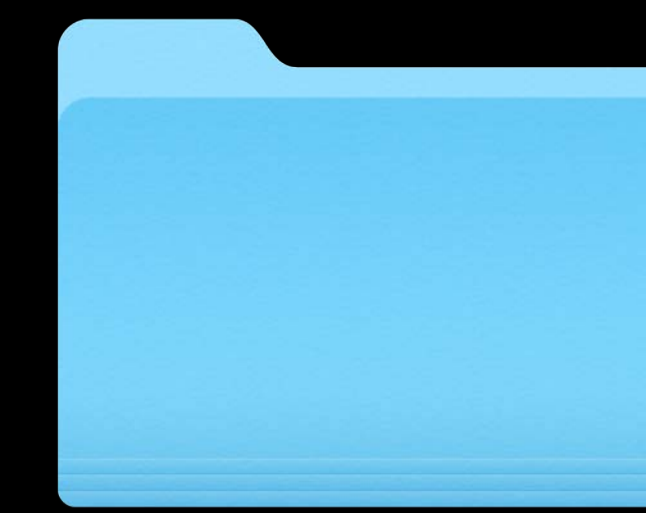
Manage







Profile Tab



Profile Tab



ProfileViewController.swift



ProfileViewController.swift



ProfilePhotoView.swift



ProfilePhotoView.swift

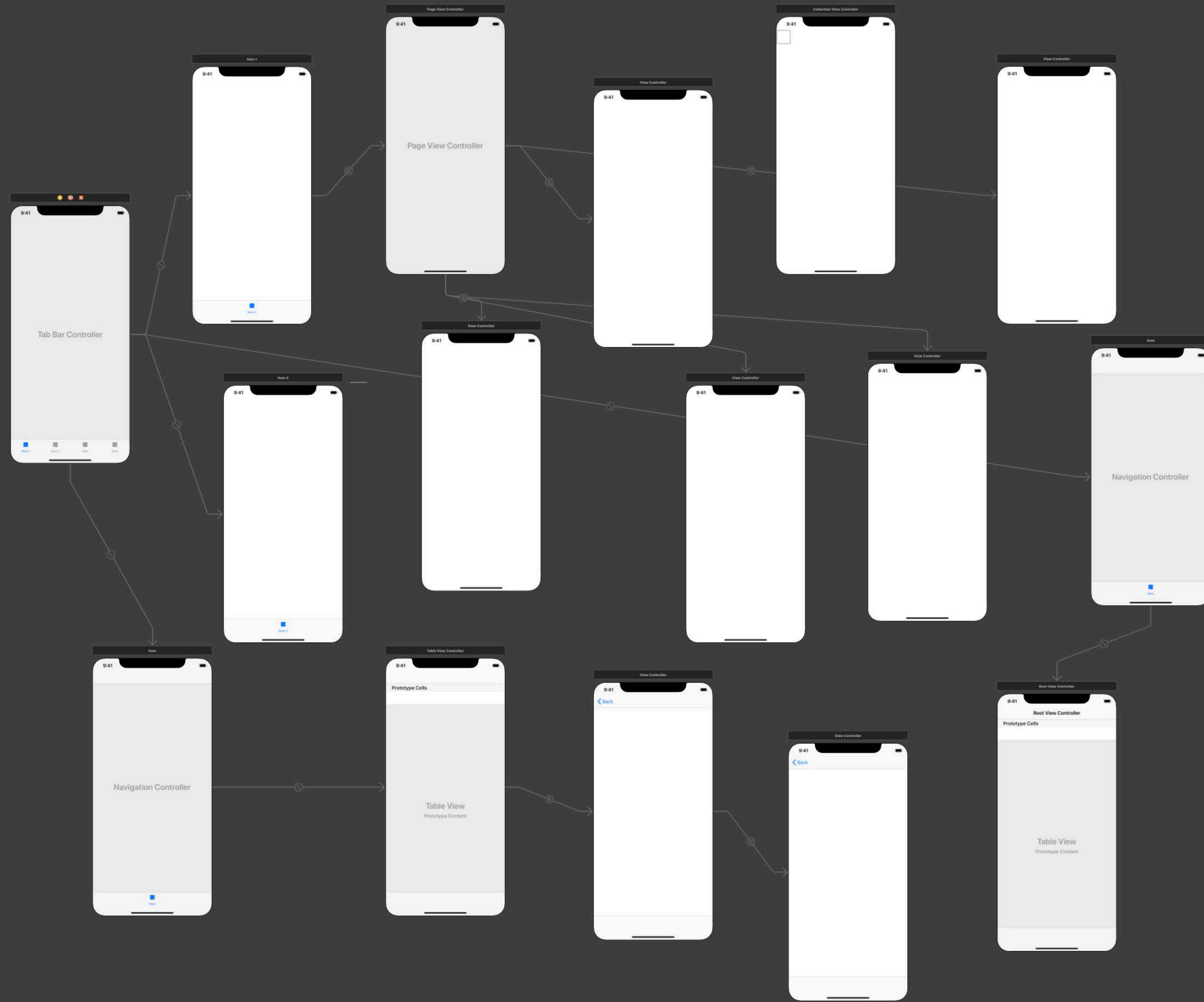


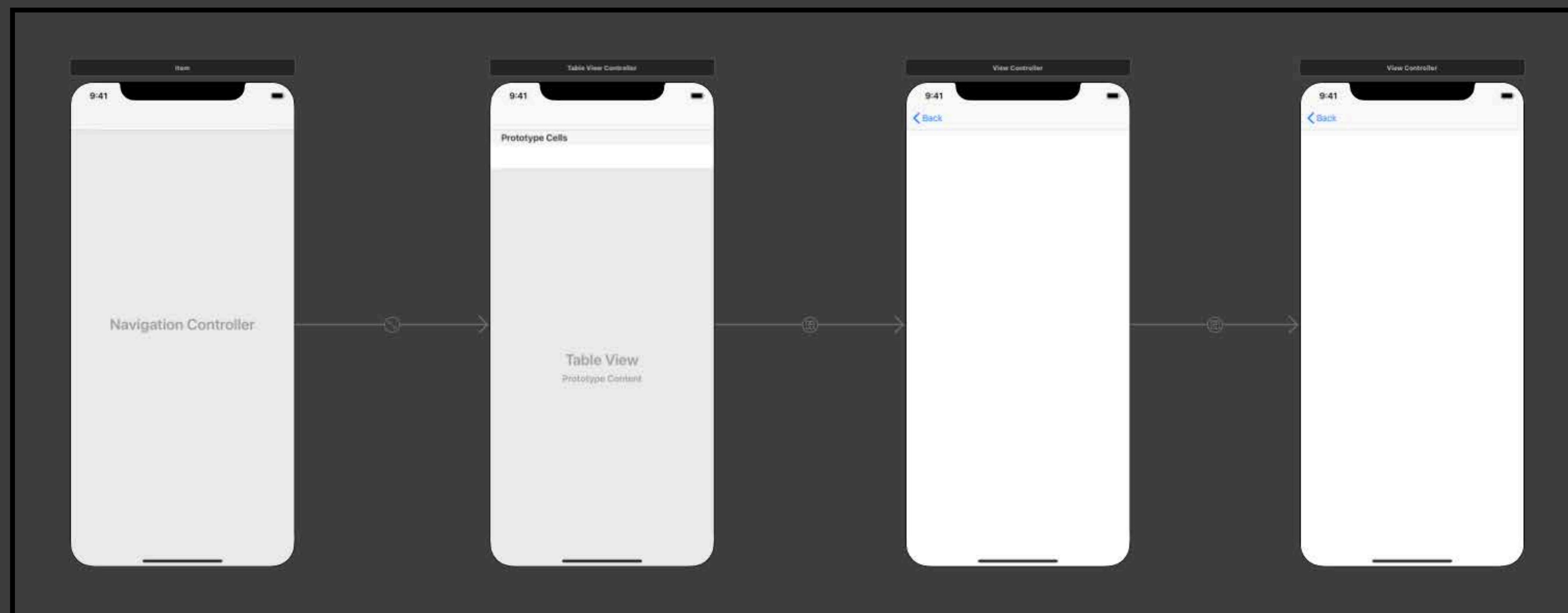
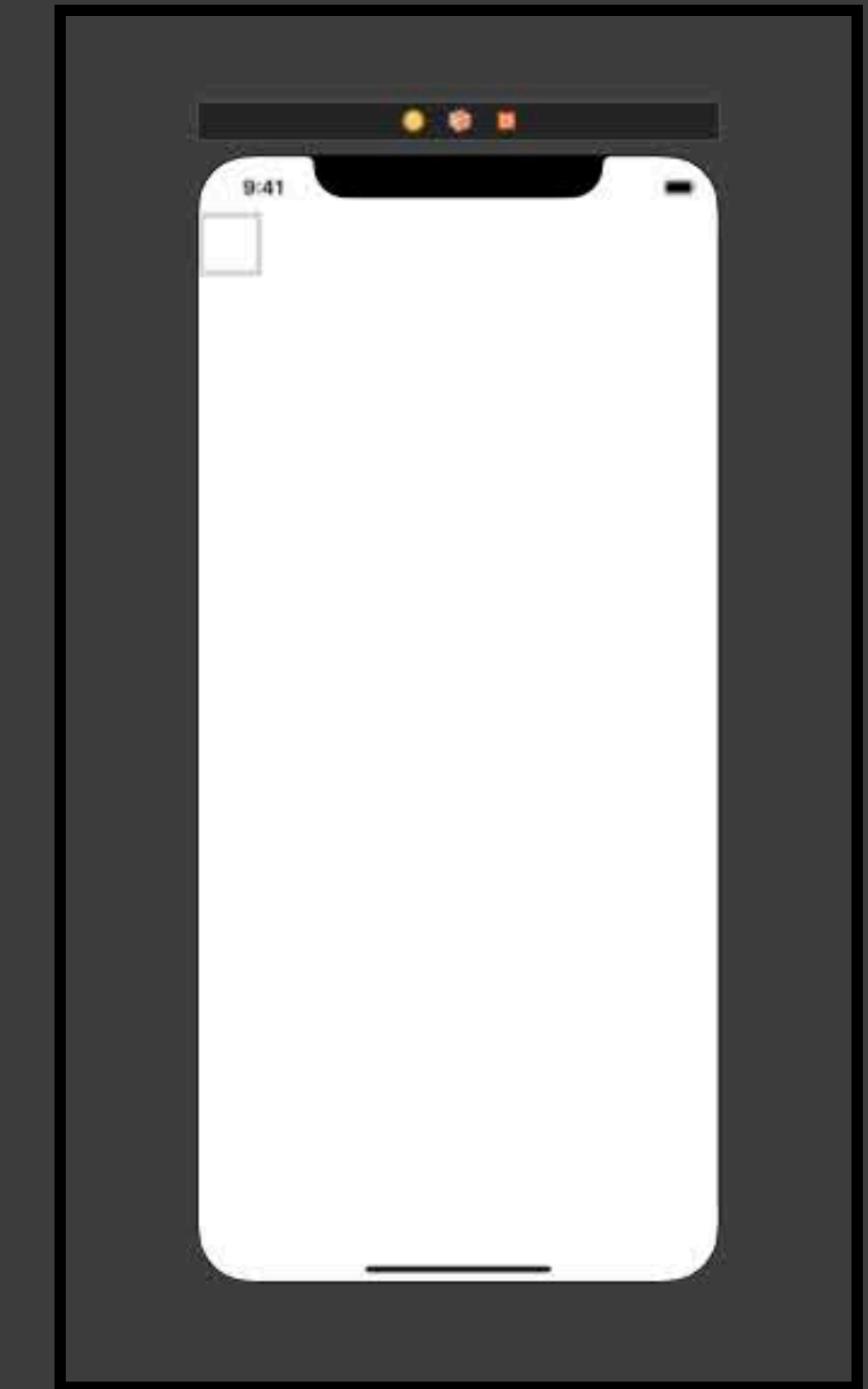
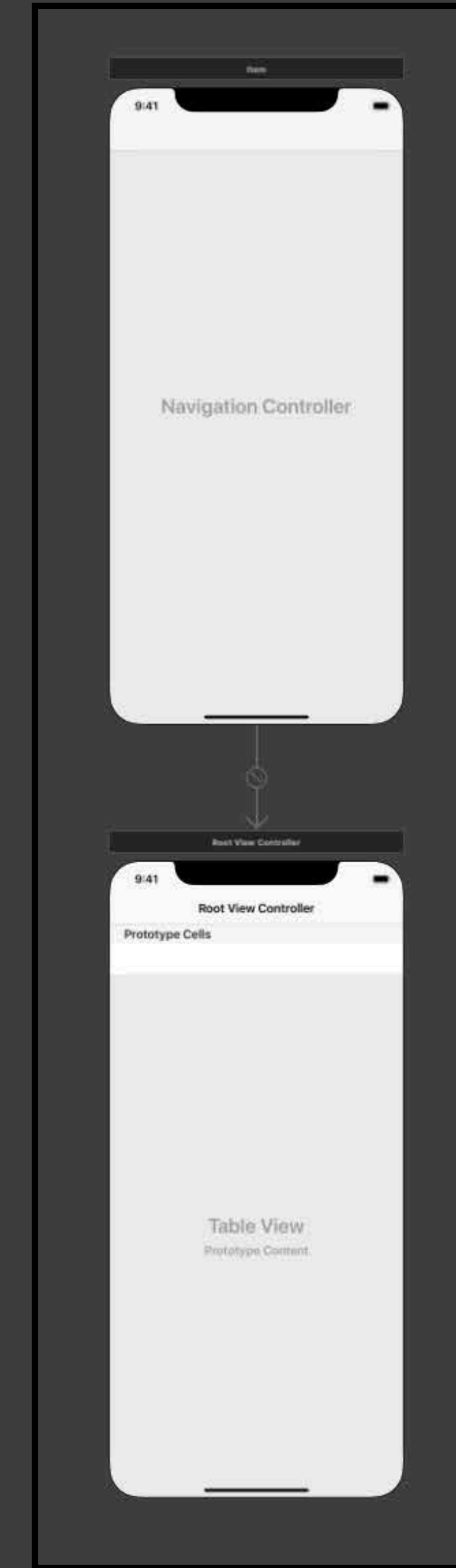
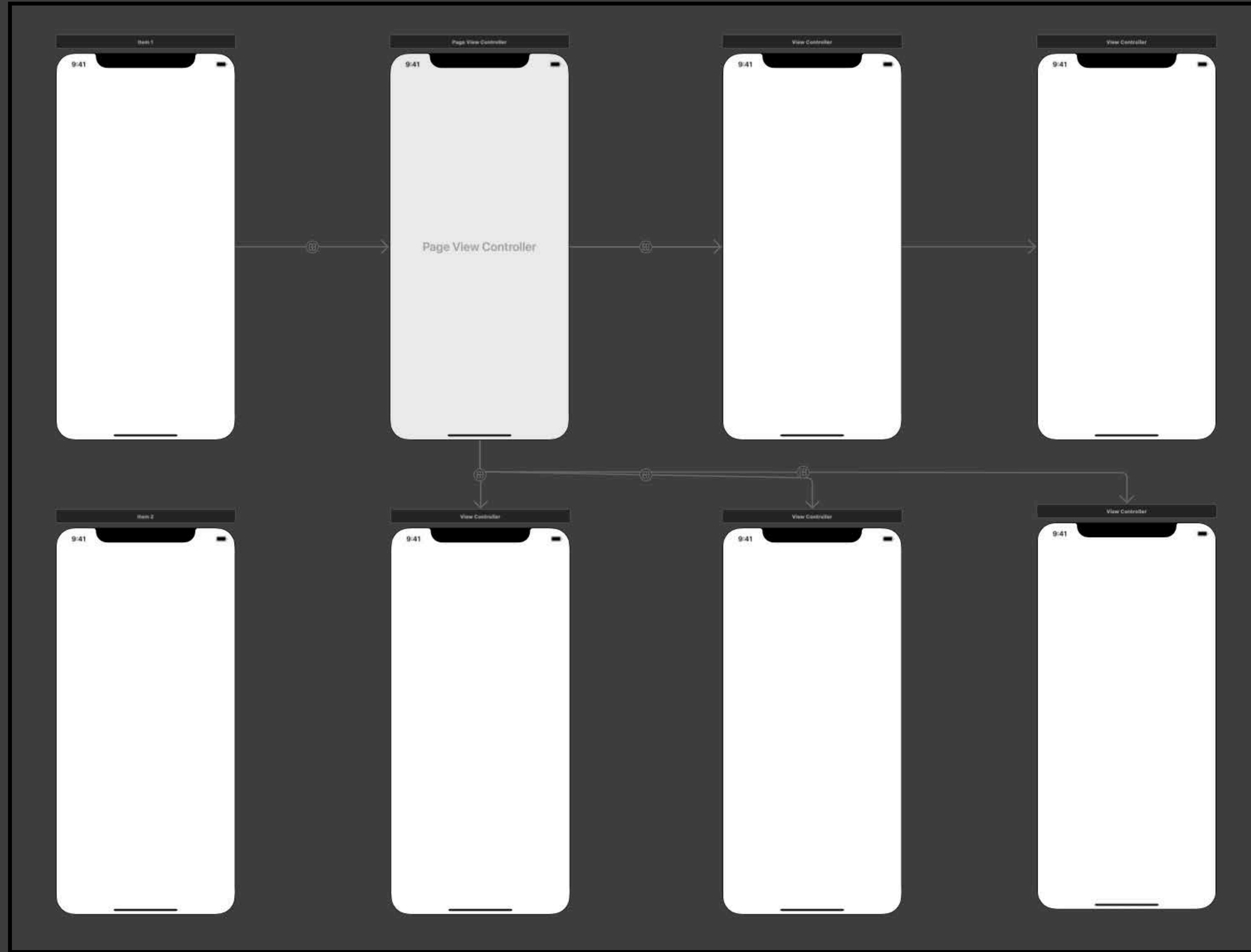
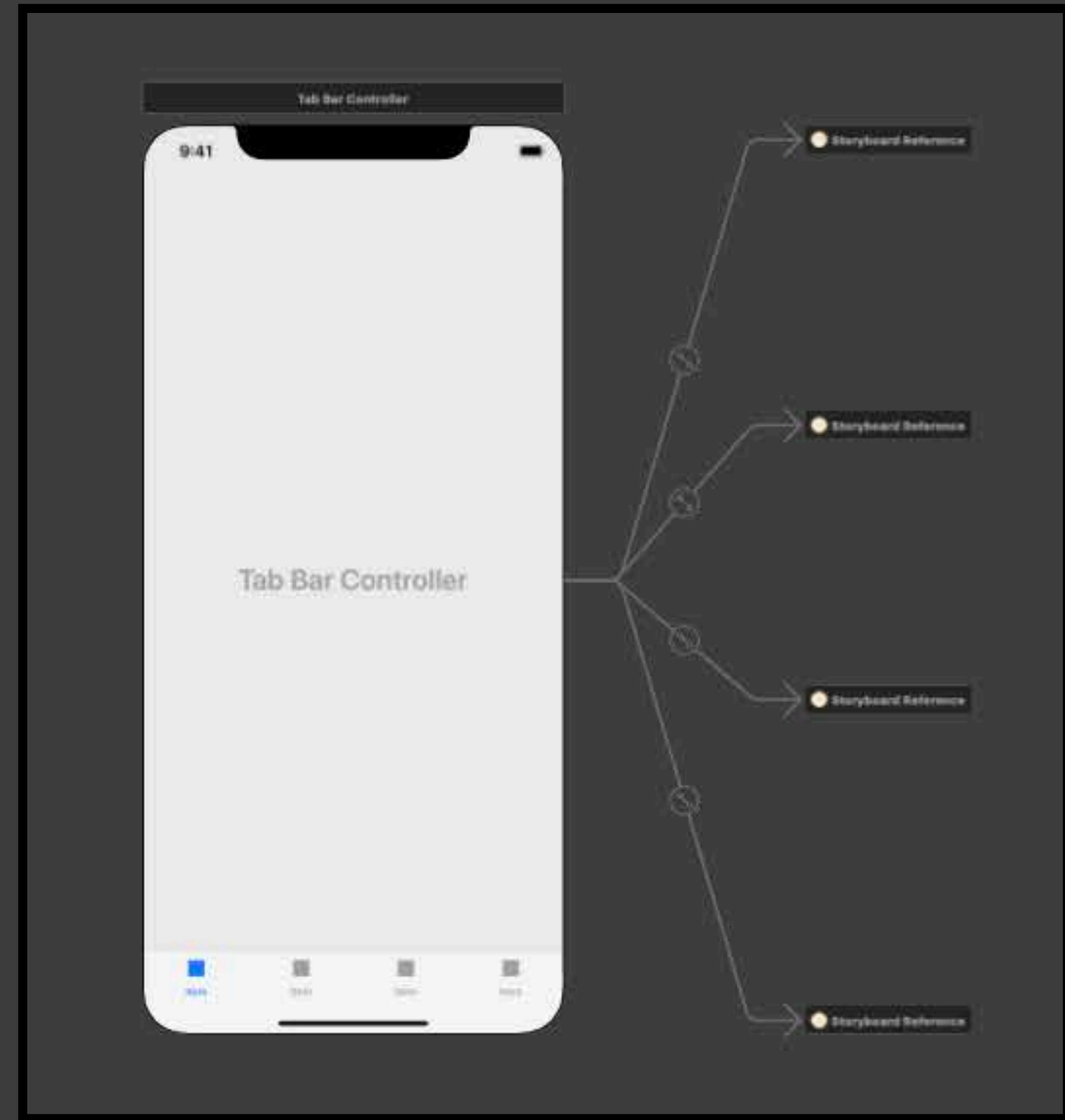
Profile.storyboard



Profile.storyboard

# Storyboards





Modern



**Buildtime (3)** Runtime

iGateway project 3 issues

- Validate Project Settings
  - Update to recommended settings iGateway.xcodeproj
- Localization
  - Enable Base Internationalization iGateway.xcodeproj
  - Migrate "English.lproj" (Deprecated) iGateway.xcodeproj

**Deployment Target**

iOS Deployment Target 4.3

**Configurations**

Name
Debug
Release

Use Release

**Localizations**

Localization
English, deprecated — Development Language
English

Use Base Internationalization

**Build Phases**

- Target 'iGateway' - Remove Unnecessary Build Files  
The iGateway target may contain unnecessary build files, such as xcconfig files or info plist files. This will remove those files from the target's build phases.

**Build Settings**

- Target 'iGateway' - Update iOS Deployment Target  
An iOS Deployment Target earlier than 8.0 is not supported by this version of Xcode. This will update the value for Target 'iGateway' to '8.0'.
- Target 'iGateway' - Adopt "Product Bundle Identifier" build setting  
Target 'iGateway' will have its Info.plist's CFBundleIdentifier key set to "\$(PRODUCT\_BUNDLE\_IDENTIFIER)", and the "PRODUCT\_BUNDLE\_IDENTIFIER" build setting will be set to
- Target 'iGateway' - Enable Weak References in Manual Retain Release  
Enabling weak references in manual retain release is recommended. This will set the CLANG\_ENABLE\_OBJC\_WEAK setting to "YES" for the "Debug" configuration.
- Target 'iGateway' - Enable Weak References in Manual Retain Release  
Enabling weak references in manual retain release is recommended. This will set the CLANG\_ENABLE\_OBJC\_WEAK setting to "YES" for the "Release" configuration.
- Project 'iGateway' - Update iOS Deployment Target  
An iOS Deployment Target earlier than 8.0 is not supported by this version of Xcode. This will update the value for Project 'iGateway' to '8.0'.
- Project 'iGateway' - Enable Recommended Warnings  
This will enable the following recommended compiler warnings:
  - Block Capture of Autoreleasing
  - Duplicate Method Definitions
  - Empty Loop Bodies
  - Enable Strict Checking of objc\_msgSend Calls
  - Implicit Boolean Conversions
  - Implicit Constant Conversions
  - Implicit Conversion to 32 Bit Type
  - Implicit Enum Conversions
  - Implicit Integer to Pointer Conversions
  - Implicit Non-Literal Null Conversions

Cancel Perform Changes

3

- delegate.swift
- ViewController.swift M
- Storyboard.storyboard M
- Assets.xcassets
- MainScreen.storyboard
- Info.plist
- AppDelegate.swift
- ViewControllerTests.swift

```

1 //
2 // ViewController.swift
3 // test
4 //
5 // Created by Joshua Tidsbury on 6/2/19.
6 // Copyright © 2019 Joshua Tidsbury. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16     }
17
18 }
19
20

```

**Shared Project Settings:**

Build System: New Build System (Default)

---

**Per-User Project Settings:**

Build System: Use Shared Setting

Derived Data: Default Location

/Users/jtidsbury/Libr.../Xcode/DerivedData

Advanced...

Issues:  Show live issues for source code  
 Show issues for active scheme only  
 Show all issues

Done

**Identity and Type**

Name: ViewController

Type: Default - Swift

---

Location: Relative to Group

ViewController

Full Path: /Users/jtidsbury/test/test/ViewController

---

**On Demand Resource Tags**

Only resources are taggable

---

**Target Membership**

test

testTests

---

**Text Settings**

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: 4

Tab

Wrap lines

```

1 //
2 // ViewController.swift
3 // test
4 //
5 // Created by Joshua Tidsbury on 6/2/19.
6 // Copyright © 2019 Joshua Tidsbury. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16     }
17
18 }
19
20

```

Shared Project Settings:

Build System: **New Build System (Default)**

---

Per-User Project Settings:

Build System: Use Shared Setting

Derived Data: Default Location

/Users/jtidsbury/Libr.../Xcode/DerivedData

Advanced...

Issues:  Show live issues for source code  
 Show issues for active scheme only  
 Show all issues

Done

Identity and Type

Name: ViewController

Type: Default - Swift

Location: Relative to Group  
ViewController

Full Path: /Users/jtidsbury/test/test/ViewController

---

On Demand Resource Tags

Only resources are taggable

---

Target Membership

test

testTests

---

Text Settings

Text Encoding: No Explicit Encoding

Line Endings: No Explicit Line Endings

Indent Using: Spaces

Widths: 4  
Tab

Wrap lines





**Organize**

# Organize

Functional organization with groups

# Organize

Functional organization with groups

Mirror project structure and file structure



# Organize

Functional organization with groups

Mirror project structure and file structure

Break apart large storyboards

# Organize

Functional organization with groups

Mirror project structure and file structure

Break apart large storyboards

Modernize your project file

# Organize

Functional organization with groups

Mirror project structure and file structure

Break apart large storyboards

Modernize your project file

Throw away code scraps

# Organize

Functional organization with groups

Mirror project structure and file structure

Break apart large storyboards

Modernize your project file

Throw away code scraps

Address the root cause of warnings

Organize

**Track**

Document

Test

Analyze

Evaluate

Decouple

Manage



No Selection

Desktop Search

- Recents
- Applications
- Desktop
- Documents
- Downloads
- Remote Disc
- Network

Source Control:  Create Git repository on my Mac  
Xcode will place your project under source control

New Folder Options Cancel Create

Cancel Previous Finish

NO EDITOR



No Selection

Desktop Search

- Recents
- Applications
- Desktop
- Documents
- Downloads
- Remote Disc
- Network

Source Control:  Create Git repository on my Mac  
 Xcode will place your project under source control

New Folder Options Cancel Create

Cancel Previous Finish

NO EDITOR









**Track**

# Track

Use source control

# Track

Use source control

Keep commits small and isolated

# Track

Use source control

Keep commits small and isolated

Write useful commit messages

# Track

Use source control

Keep commits small and isolated

Write useful commit messages

Utilize branches for bug and feature work

Organize

Track

**Document**

Test

Analyze

Evaluate

Decouple

Manage



“I don't need comments,  
my code is self-documenting”





```
// The permanent identifier for this application when interacting
// with the CMS, provided by the vendor of the CMS.
let id = "2ADA155F-1529-4D2D-98C4-0E4BD06940E8"
```

```
// The permanent identifier for this application when interacting
// with the CMS, provided by the vendor of the CMS.
let cmsApplicationIdentifier = "2ADA155F-1529-4D2D-98C4-0E4BD06940E8"
```

myApp: Ready | Today at 5:05 PM

myApp > myApp > ViewController.swift > ViewController

myApp 1 Breakpoint  
Ex Undefined Behavior (Runtime Iss...

1 //  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21

Swift Objective-C REST JS

UIKit > View Controllers > UIViewController

App Frameworks

- AppKit
- Bundle Resources
- Foundation
- Swift
- SwiftUI
- TVML
- TVMLKit
- TVUIKit
- UIKit
- WatchKit

App Services

- Accounts
- AddressBook
- AddressBookUI
- AdSupport
- ApplicationServices
- BackgroundTasks
- Business Chat
- CallKit
- CarPlay
- ClassKit
- ClockKit
- CloudKit
- Combine
- Contacts
- ContactsUI
- Core Data
- Core Foundation
- Core Location
- Core ML
- Core Motion
- Core Spotlight
- Core Text
- Create ML
- DeviceCheck
- EventKit
- EventKitUI
- FileProvider
- FileProviderUI
- HealthKit
- HomeKit
- iAd

Filter

Search documentation

# Class UIViewController

An object that manages a view hierarchy for your UIKit app.

## Declaration

```
class UIViewController : UIResponder
```

## Overview

The `UIViewController` class defines the shared behavior that is common to all view controllers. You rarely create instances of the `UIViewController` class directly. Instead, you subclass `UIViewController` and add the methods and properties needed to manage the view controller's view hierarchy.

A view controller's main responsibilities include the following:

- Updating the contents of the views, usually in response to changes to the underlying data.
- Responding to user interactions with views.
- Resizing views and managing the layout of the overall interface.
- Coordinating with other objects—including other view controllers—in your app.

A view controller is tightly bound to the views it manages and takes part in handling events in its view hierarchy. Specifically, view controllers are `UIResponder` objects and are inserted into the responder chain between the view controller's root view and that view's superview, which typically belongs to a different view controller. If none of the view controller's views handle an event, the view controller has the option of handling the event or passing it along to the superview.

View controllers are rarely used in isolation. Instead, you often use multiple view

### Language

Swift | Objective-C

### SDKs

iOS 2.0+  
tvOS 9.0+

### Framework

UIKit

### On This Page

- [Declaration](#)
- [Overview](#)
- [Topics](#)
- [Relationships](#)

### Quick Help

#### Summary

An object that manages a view hierarchy for your UIKit app.

#### Declaration

```
class UIViewController : UIResponder
```

#### Discussion

The `UIViewController` class defines the shared behavior that is common to all view controllers. You rarely create instances of the `UIViewController` class directly. Instead, you subclass `UIViewController` and add the methods and properties needed to manage the view controller's view hierarchy.

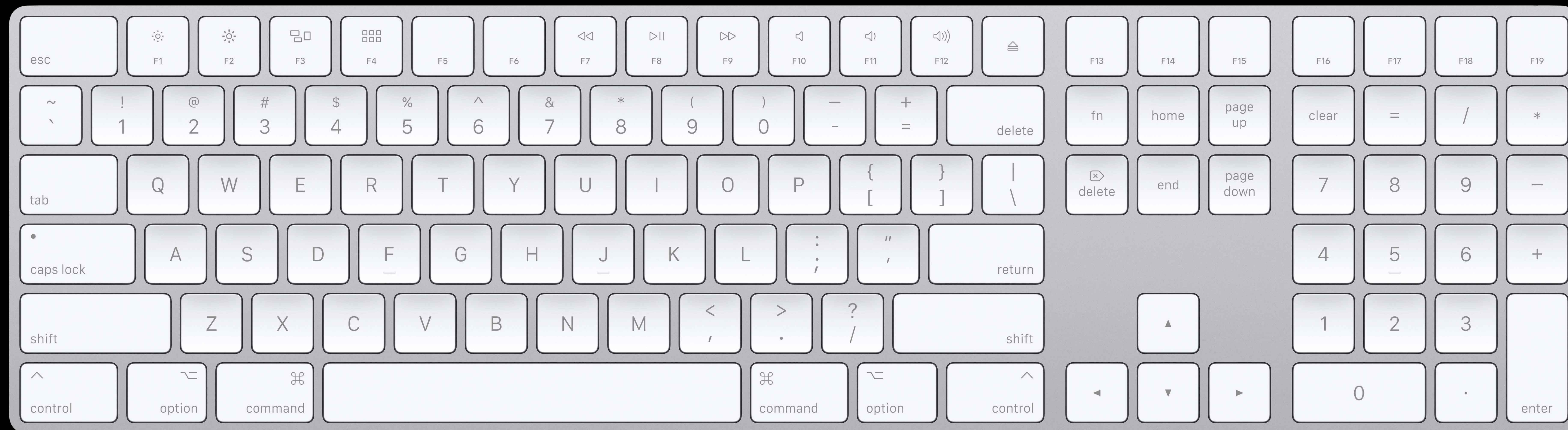
A view controller's main responsibilities include the following:

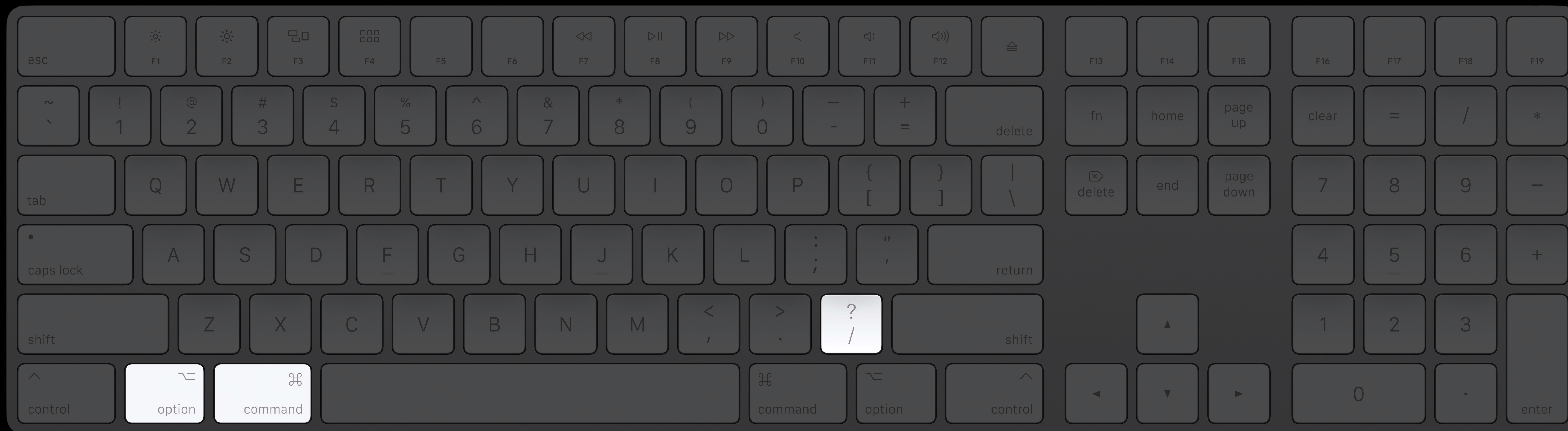
- Updating the contents of the views, usually in response to changes to the underlying data.
- Responding to user interactions with views.
- Resizing views and managing the layout of the overall interface.
- Coordinating with other objects—including other view controllers—in your app.

A view controller is tightly bound to the views it manages and takes part in handling events in its view hierarchy. Specifically, view controllers are `UIResponder` objects and are inserted into the responder chain between the view controller's root view and that view's superview, which typically belongs to a different view controller. If none of the view controller's views handle an event, the view controller has the option of handling the event or passing it along to the superview.

View controllers are rarely used in isolation. Instead, you often use multiple view controllers, each of which owns a portion of your app's user interface. For example, one view controller might display a table of items while a different view controller displays the selected item from that table. Usually, only the views from one view controller are visible at a time. A view controller may present a different view controller to display a new set of views, or it may act as a container for other view controllers' content and animate views however it wants.

[Open in Developer Documentation](#)







The screenshot shows the Xcode IDE interface. At the top, the status bar indicates 'myApp: Ready | Today at 4:53 PM'. The breadcrumb navigation shows the current file is 'ViewController.swift' with 'No Selection'. The left sidebar displays a breakpoint for 'myApp 1 Breakpoint' with an 'Undefined Behavior (Runtime Iss...' issue. The main editor area shows the following Swift code:

```
1 //
2 // ViewController.swift
3 // myApp
4 //
5 // Created by Johnny Appleseed on 6/3/19.
6 // Copyright © 2019 Apple Inc. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
```

A tooltip is displayed over the `UIViewController` class name, containing the following information:

- Summary**  
An object that manages a view hierarchy for your UIKit app.
- Declaration**  
`class UIViewController : UIResponder`
- Discussion**  
The `UIViewController` class defines the shared behavior that is common to all view controllers. You rarely create instances of the `UIViewController` class directly. Instead, you subclass `UIViewController` and add the methods and properties needed to manage the view controller's view hierarchy.  
A view controller's main responsibilities include the following:
  - Updating the contents of the views, usually in response to changes to the underlying data.
  - Responding to user interactions with views.
  - Resizing views and managing the layout of the overall interface.
  - Coordinating with other objects—including other view controllers—in your app.A view controller is tightly bound to the views it manages and takes part in handling events in its view hierarchy. Specifically, view controllers are `UIResponder` objects and are inserted into the responder chain between the view controller's root view and that view's superview, which typically belongs to a different view controller. If none of the view controller's views handle an event, the view controller has the option of handling the event or passing it along to the superview.  
View controllers are rarely used in isolation. Instead, you often use multiple view controllers, each of which owns a portion of your app's user interface. For example, one view controller might display a table of items while a different view controller displays the selected item from that table. Usually, only the views from one view controller are visible at a time. A view controller may present a different view controller to display a new set of views, or it may act as a container for other view controllers' content and animate views however it wants.

[Open in Developer Documentation](#)

**Document**

# Document

Comments are critical for future understanding

# Document

Comments are critical for future understanding

Good comments provide background and reasoning

# Document

Comments are critical for future understanding

Good comments provide background and reasoning

Use descriptive variable and constant names

# Document

Comments are critical for future understanding

Good comments provide background and reasoning

Use descriptive variable and constant names

Include documentation

Organize

Track

Document

**Test**

Analyze

Evaluate

Decouple

Manage

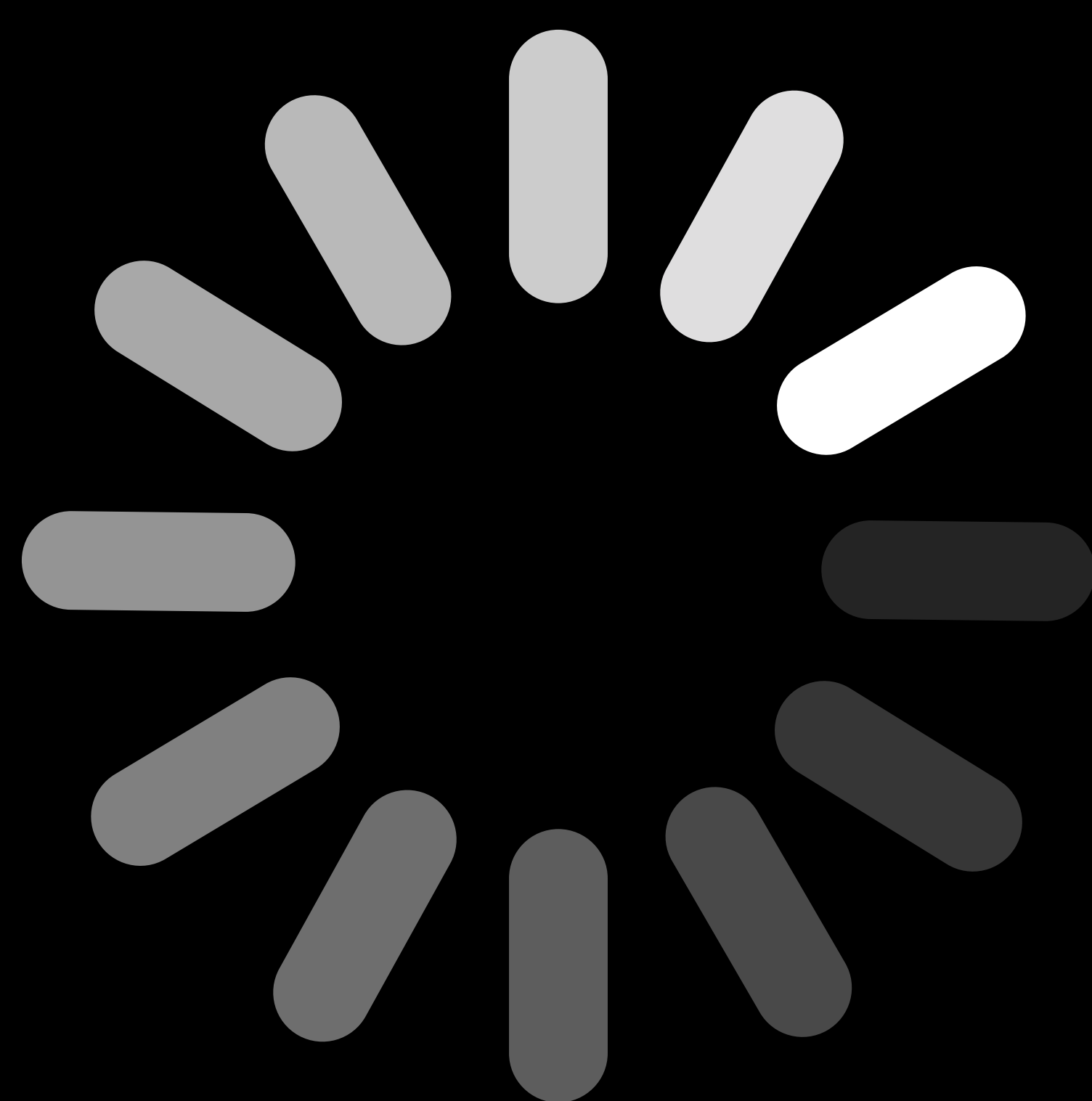




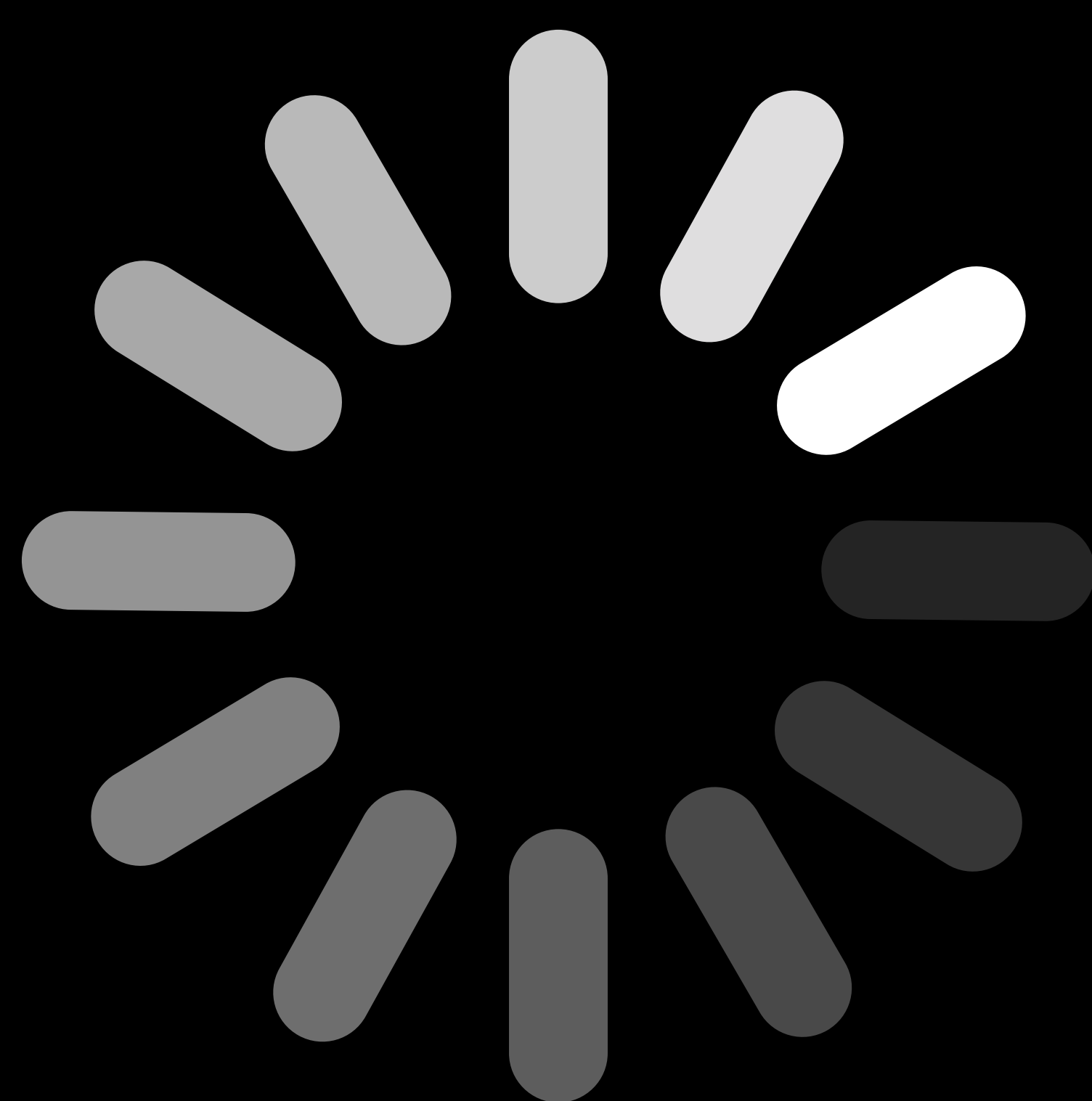


“And while you’re doing that, you might as well add a unit test to make sure that the round trip between the Struct and the dictionary representation keeps working.”

“And while you’re doing that, you might as well add a unit test to make sure that the round trip between the Struct and the dictionary representation keeps working.”















# Unit Tests

# Unit Tests

Write unit tests

# Unit Tests

Write unit tests

Run unit tests before committing code

# Unit Tests

Write unit tests

Run unit tests before committing code

Build a foundation for continuous integration

Organize

Track

Document

Test

**Analyze**

Evaluate

Decouple

Manage

## INSTALLED APPS

Name	Version	Identifier
 Travel	1	com.apple.dt.Trav



## DEVICE CONDITIONS

Condition

Profile  100% packet loss

- Very poor network
- Edge Network - poor
- Edge Network - average
- Edge Network - good
- Edge Network - best
- 2G Network - poor
- 2G Network - better
- 3G Network - average
- 3G Network - good
- 3G Network - best
- LTE Network
- WiFi Network
- WiFi Network (802.11ac)
- DSL Network
- High Latency DNS

+

```

1 //
2 // ViewController.swift
3 // myApp
4 //
5 // Created by Johnny Apple
6 // Copyright © 2019 Apple
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional
16     }
17
18 }
19
20
21

```

myApp > iPhone XR

- Build (3 targets)
- Run (Debug)**
- Test (Debug)
- Profile (Release)
- Analyze (Debug)
- Archive (Release)

Info	Arguments	Options	Diagnostics
Runtime Sanitization <small>Requires recompilation</small>	<input checked="" type="checkbox"/> Address Sanitizer	<input type="checkbox"/> Detect use of stack after return	
	<input type="checkbox"/> Thread Sanitizer ⓘ	<input type="checkbox"/> Undefined Behavior Sanitizer ↕	
Runtime API Checking	<input type="checkbox"/> Main Thread Checker ↕		
Memory Management	<input type="checkbox"/> Malloc Scribble		
	<input type="checkbox"/> Malloc Guard Edges ⓘ		
	<input type="checkbox"/> Guard Malloc ⓘ		
	<input type="checkbox"/> Zombie Objects		
	<input type="checkbox"/> Malloc Stack ⓘ		
	Live Allocations Only ▾		
	<input type="checkbox"/> Memory Graph on Resource Exception ⓘ		

Shared

Identity and Target

Name

Type

Location

Full Path

On Demand Resources

Only resources

Target Membership

- myApp
- myApp
- myApp

Text Settings

Text Encoding

Line Ending

Indent Using

Width



```

1 //
2 // ViewController.swift
3 // myApp
4 //
5 // Created by Johnny Apple
6 // Copyright © 2019 Apple
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional
16     }
17
18 }
19 }
20
21

```

- ▶ Build  
3 targets
- ▶ Run  
Debug
- ▶ Test  
Debug
- ▶ Profile  
Release
- ▶ Analyze  
Debug
- ▶ Archive  
Release

Info Arguments Options Diagnostics

Runtime Sanitization  
Requires recompilation

- Address Sanitizer
  - Detect use of stack after return
  - Thread Sanitizer ⓘ
  - Undefined Behavior Sanitizer ↕

Runtime API Checking  Main Thread Checker ↕

Memory Management

- Malloc Scribble
- Malloc Guard Edges ⓘ
- Guard Malloc ⓘ
- Zombie Objects
- Malloc Stack ⓘ
- Live Allocations Only ▾
- Memory Graph on Resource Exception ⓘ

?

Identity and T

Name

Type

Location

Full Pat

On Demand R

Only resou

Target Membe

- myApp
- myApp
- myApp

Text Settings

Text Encodin

Line Ending

Indent Usin

Width

```
1 //
2 // ViewController.swift
3 // myApp
4 //
5 // Created by Johnny Apple
6 // Copyright © 2019 Apple
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional
16     }
17
18 }
19 }
20
21
```

- ▶ Build  
3 targets
- ▶ Run  
Debug
- ▶ Test  
Debug
- ▶ Profile  
Release
- ▶ Analyze  
Debug
- ▶ Archive  
Release

Info Arguments Options Diagnostics

- Runtime Sanitization  
Requires recompilation
  - Address Sanitizer ⓘ
  - Detect use of stack after return
  - Thread Sanitizer ⓘ
  - Undefined Behavior Sanitizer ⓘ
- Runtime API Checking
  - Main Thread Checker ⓘ
- Memory Management
  - Malloc Scribble ⓘ
  - Malloc Guard Edges ⓘ
  - Guard Malloc ⓘ
  - Zombie Objects
  - Malloc Stack ⓘ
  - Live Allocations Only
  - Memory Graph on Resource Exception ⓘ

```

1 //
2 // ViewController.swift
3 // myApp
4 //
5 // Created by Johnny Apple
6 // Copyright © 2019 Apple
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional
16     }
17
18 }
19
20
21

```

- ▶ Build  
3 targets
- ▶ Run  
Debug
- ▶ Test  
Debug
- ▶ Profile  
Release
- ▶ Analyze  
Debug
- ▶ Archive  
Release

Info Arguments Options Diagnostics

- Runtime Sanitization  
Requires recompilation
  - Address Sanitizer
  - Detect use of stack after return
  - Thread Sanitizer
  - Undefined Behavior Sanitizer
- Runtime API Checking
  - Main Thread Checker
- Memory Management
  - Malloc Scribble
  - Malloc Guard Edges
  - Guard Malloc
  - Zombie Objects
  - Malloc Stack
  - Live Allocations Only
  - Memory Graph on Resource Exception

```

1 //
2 // ViewController.swift
3 // myApp
4 //
5 // Created by Johnny Apple
6 // Copyright © 2019 Apple
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional
16     }
17
18 }
19
20
21

```

- ▶ Build  
3 targets
- ▶ Run  
Debug
- ▶ Test  
Debug
- ▶ Profile  
Release
- ▶ Analyze  
Debug
- ▶ Archive  
Release

Info Arguments Options Diagnostics

Runtime Sanitization  
Requires recompilation

- Address Sanitizer
- Detect use of stack after return
- Thread Sanitizer
- Undefined Behavior Sanitizer

Runtime API Checking  Main Thread Checker

Memory Management

- Malloc Scribble
- Malloc Guard Edges
- Guard Malloc
- Zombie Objects
- Malloc Stack
- Live Allocations Only
- Memory Graph on Resource Exception

Identity and T

Name

Type

Location

Full Pat

On Demand R

Only resou

Target Membe

- myApp
- myApp
- myApp

Text Settings

Text Encodin

Line Ending

Indent Usin

Width

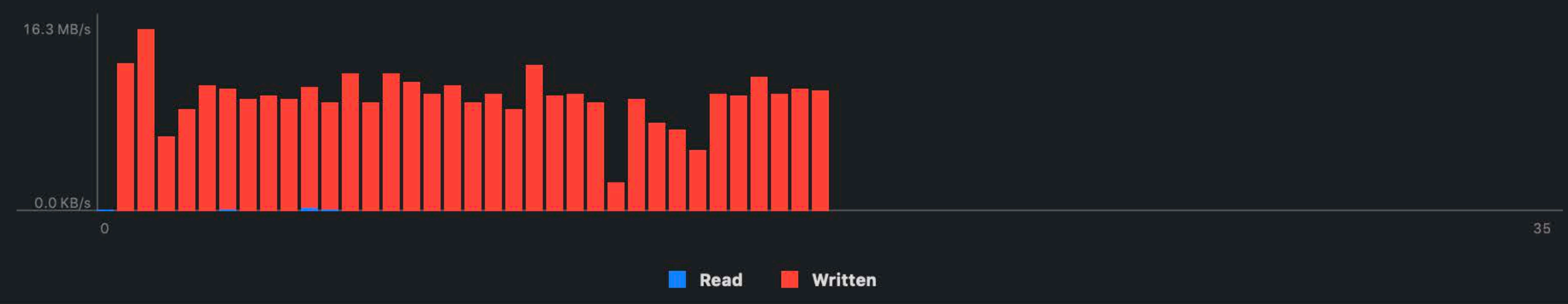
- myApp PID 22284
- CPU 0%
- Memory 51.5 MB
- Disk 11 MB/s**
- Network Zero KB/s

### Disk

Profile in Instruments

Reading		Writing	
0.0 KB/s	0.6 MB	10.9 MB/s	0.3 GB
Per Second	Total	Per Second	Total

### Reading and Writing Rates



### Total Read and Write



### Files

Descriptor	Type	Device	Size/Offset	Inode	Status	Path
1u	CHR	268,435,457	0	1,645	Open	/dev/ttys001
0r	CHR	50,331,650	0	310	Open	/dev/null
2u	CHR	268,435,457	428	1,645	Open	/dev/ttys001
3r	REG	0	0	8,615,061,091	Open	/Users/johnnyappleseed/Library/Developer/CoreSimulator/Devices/161C3B23-45FE-46D0-96...

Instruments

Run 1 of 1 | 00:00:27

Track Filter: All Tracks Instruments CPUs Threads Duplicate

Time Profiler Instrument CPU Usage

Points of Interest Instrument Points

Thermal State Instrument Current Nominal

Time Profiler Profile Root

Weight	Self Weight	Symbol Name
331.00 ms	100.0%	0 s ▼myApp (22439)
283.00 ms	85.4%	0 s ▼Main Thread 0xe07e04
158.00 ms	47.7%	0 s ▼_dyld_start dyld
158.00 ms	47.7%	0 s ▼dyldbootstrap::start(macho_header const*, int, char const**, long, macho_header const*, unsigned long*) dyld
158.00 ms	47.7%	0 s ▼dyld::_main(macho_header const*, unsigned long, int, char const**, char const**, char const**, unsigned long*) dyld
158.00 ms	47.7%	0 s ▼dyld::useSimulatorDyld(int, macho_header const*, char const*, int, char const**, char const**, char const**, unsigned long*, unsigned long*) dyld
157.00 ms	47.4%	0 s ▼0x1089441cc
84.00 ms	25.3%	0 s ▼0x108948f61
84.00 ms	25.3%	0 s ▼0x108947c41
46.00 ms	13.8%	0 s ▼0x10894fe09
42.00 ms	12.6%	0 s ▼0x108950e1a
38.00 ms	11.4%	0 s ▼0x108950e1a
37.00 ms	11.1%	0 s ▼0x108950e1a
35.00 ms	10.5%	0 s ▼0x108950e1a
26.00 ms	7.8%	0 s ▼0x108950e1a
24.00 ms	7.2%	0 s ▼0x108950e1a
17.00 ms	5.1%	0 s ▼0x108950e1a
9.00 ms	2.7%	0 s ▼0x108950e1a
7.00 ms	2.1%	0 s ▼0x108950e1a
6.00 ms	1.8%	0 s ▼0x108950e1a
5.00 ms	1.5%	0 s ▼0x108950e1a
4.00 ms	1.2%	0 s ▼0x108950e1a
3.00 ms	0.9%	0 s ▼0x108950e1a
3.00 ms	0.9%	0 s ▼0x108950e1a
2.00 ms	0.6%	0 s ▼0x108950c35
2.00 ms	0.6%	0 s ▼0x10894ad02
2.00 ms	0.6%	0 s ▼0x108945de4
2.00 ms	0.6%	0 s ▼0x1089460c5
2.00 ms	0.6%	0 s ▼0x1089499c4
2.00 ms	0.6%	0 s ▼0x10894a1f0
2.00 ms	0.6%	0 s ▼0x10894a496
1.00 ms	0.3%	0 s ▼0x10894a913
1.00 ms	0.3%	1.00 ms open dyld
1.00 ms	0.3%	0 s ▼0x10894a933
1.00 ms	0.3%	0 s ▼0x1089463b0
1.00 ms	0.3%	0 s ▼0x108953884
1.00 ms	0.3%	0 s ▼0x108957c54
1.00 ms	0.3%	0 s ▼0x108956e52
1.00 ms	0.3%	1.00 ms _mmap dyld
1.00 ms	0.3%	0 s ▼0x108950e1a
1.00 ms	0.3%	0 s ▼0x108950e1a

Heaviest Stack Trace

- 331 myApp (22439)
- 283 Main Thread 0xe07e04
- 158 \_dyld\_start
- 158 dyldbootstrap::start(macho\_header const\*, int, char const\*\*, long, macho\_header const\*, unsigned long\*) dyld
- 158 dyld::\_main(macho\_header const\*, unsigned long, int, char const\*\*, char const\*\*, char const\*\*, unsigned long\*) dyld
- 158 dyld::useSimulatorDyld(int, macho\_header const\*, char const\*, int, char const\*\*, char const\*\*, char const\*\*, unsigned long\*, unsigned long\*) dyld
- 157 0x1089441cc
- 84 0x108948f61
- 84 0x108947c41
- 46 0x10894fe09
- 42 0x108950e1a
- 38 0x108950e1a
- 37 0x108950e1a
- 35 0x108950e1a
- 26 0x108950e1a
- 24 0x108950e1a
- 17 0x108950e1a
- 9 0x108950e1a
- 7 0x108950e1a
- 6 0x108950e1a
- 5 0x108950e1a
- 4 0x108950e1a
- 3 0x108950e1a
- 3 0x108950e1a
- 2 0x108950c35
- 2 0x10894ad02
- 2 0x108945de4
- 2 0x1089460c5
- 2 0x1089499c4
- 2 0x10894a1f0
- 2 0x10894a496
- 2 0x10894a913
- 2 0x10894a933
- 2 0x1089463b0
- 2 0x10894a496
- 1 0x10894a913
- 1 open

Input Filter Involves Symbol Call Tree Call Tree Constraints Data Mining

**Analyze**

# Analyze

Simulate poor networks with Network Link Conditioner



# Analyze

Simulate poor networks with Network Link Conditioner

Use sanitizers and checkers

# Analyze

Simulate poor networks with Network Link Conditioner

Use sanitizers and checkers

Measure performance and efficiency with Debug Gauges

# Analyze

Simulate poor networks with Network Link Conditioner

Use sanitizers and checkers

Measure performance and efficiency with Debug Gauges

Investigate issues with Instruments

Organize

Track

Document

Test

Analyze

**Evaluate**

Decouple

Manage











Understand each change

Build it

Run tests

Proofread

```
let colour = ...
```

```
let colour| = ...
```

```
let colour| = ...
```

```
let color| = ...
```



```
let color = ...
```

**Evaluate**

# Evaluate

Include code review as part of your practice

# Evaluate

Include code review as part of your practice

Understand each line

# Evaluate

Include code review as part of your practice

Understand each line

Build it

# Evaluate

Include code review as part of your practice

Understand each line

Build it

Run tests

# Evaluate

Include code review as part of your practice

Understand each line

Build it

Run tests

Proofread for style, spelling, and syntax

Organize

Track

Document

Test

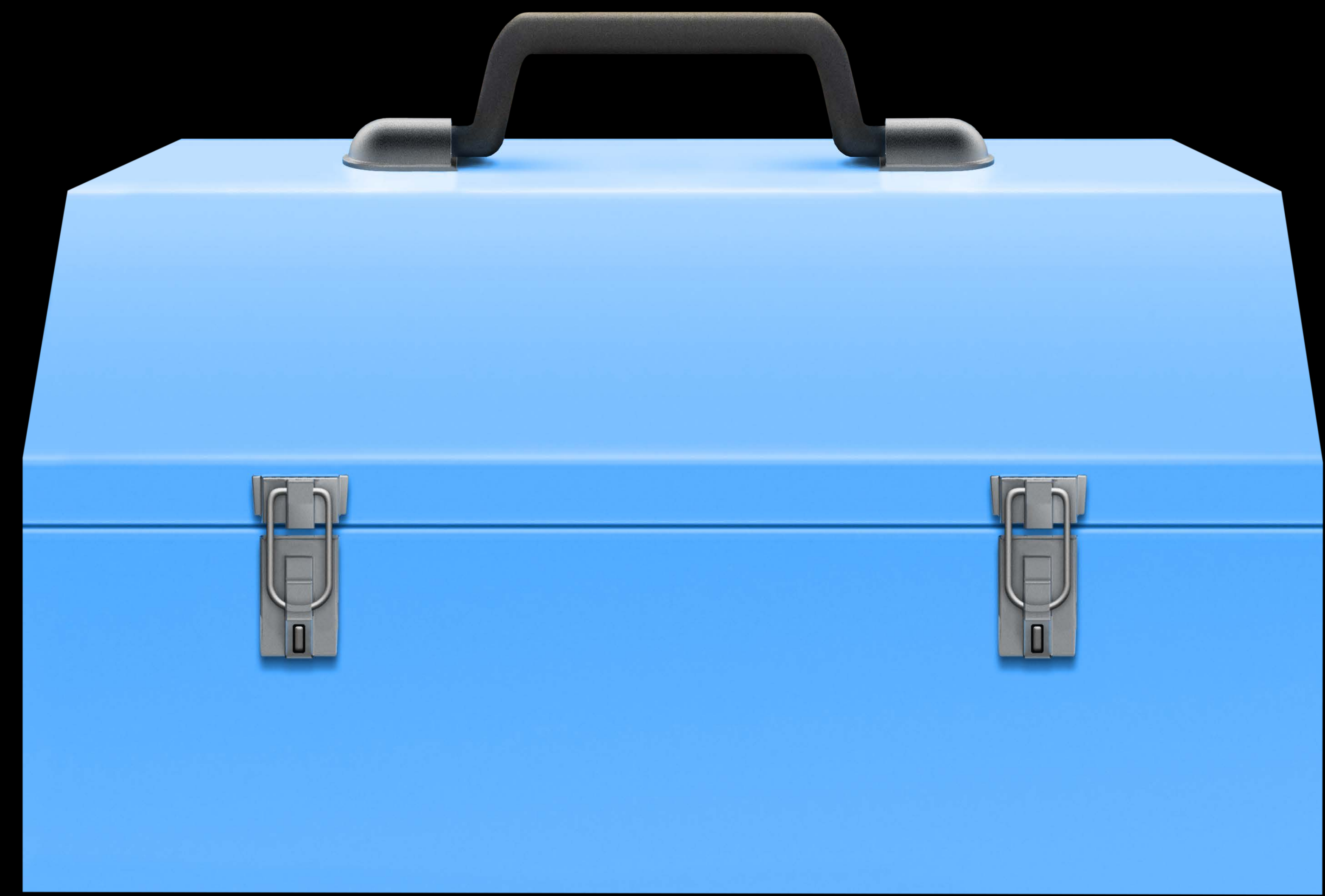
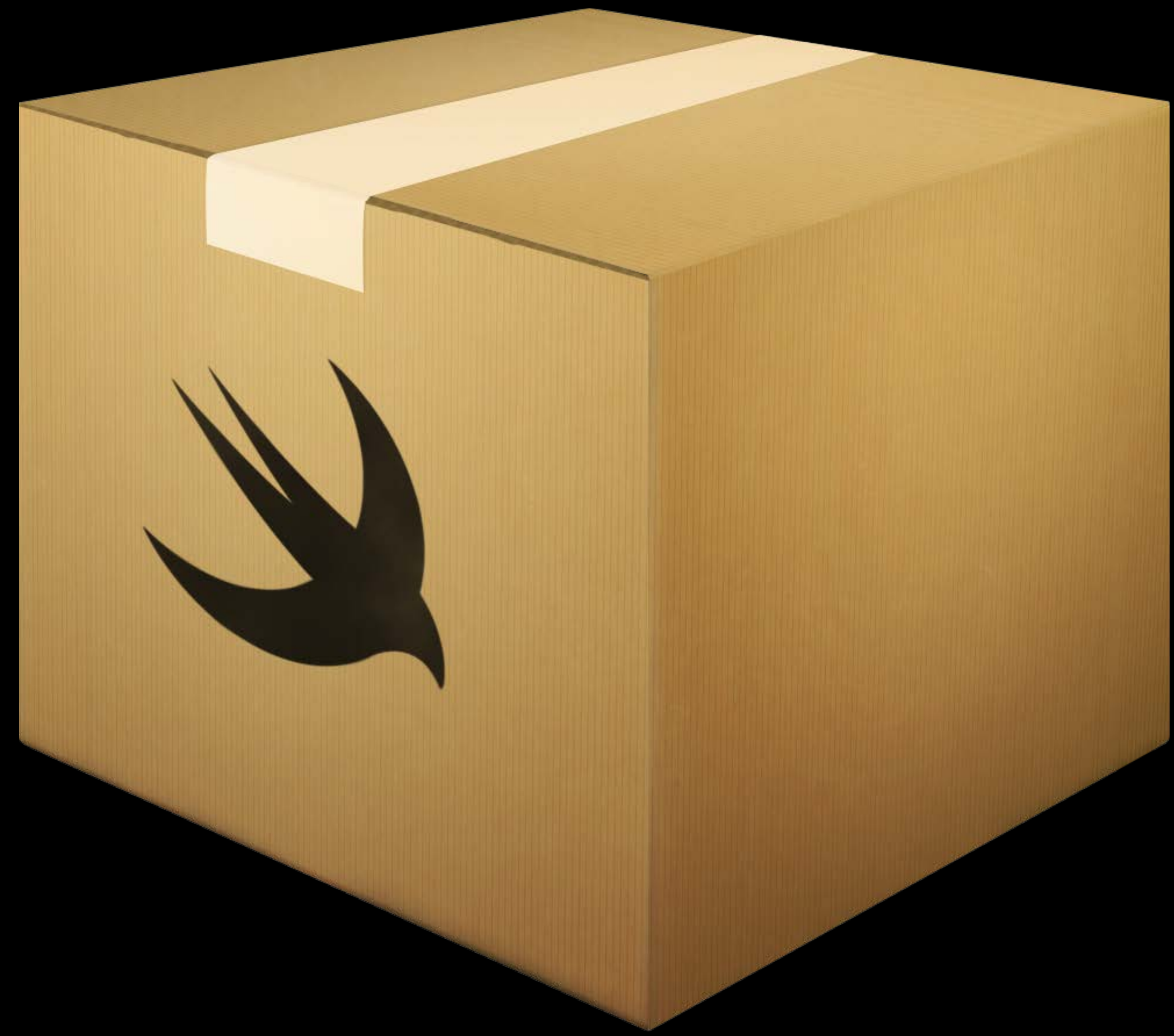
Analyze

Evaluate

**Decouple**

Manage









Bitbucket



GitHub



GitLab

```
177
178     /// Sets whether the specified `identifier` is a favorite or not. This function will kick off
179     /// any necessary work to save the information to disk, post notifications, or initiate a
180     /// sync with CloudKit asynchronously.
181     ///
182     /// Note: The `notify` and `syncToCloudKit` options will *always* cause the asynchronous
183     /// work to post a notification and/or sync to CloudKit regardless of whether the state of
184     /// the favorite is actually changing.
185     ///
186     /// - Parameters:
187     ///   - status: whether the item is a favorite or not
188     ///   - identifier: the identifier of the item
189     ///   - notify: if true, will post the `.WWDCFavoritesDidChange` notification
190     ///   - syncToCloudKit: if true, will initiate an async operation to push the state to CloudKit
191     ///   - lastModified: the date to record as the last modified date. Defaults to the current date.
192     ///     This is useful when syncing from CloudKit and we want to record a date that matches the server.
193     public func setFavorite(_ status: Bool, for identifier: Activity.Identifier, notify: Bool = true, syncToCloudKit: Bool = true, lastModified: Date = Date()) {
194         var favorite = favoritesByIdentifier[identifier]
195         var needSave = false
196
197         // If we don't have an entry for this identifier then create one now.
198         if favorite == nil {
199             favorite = FavoriteEntry(activityIdentifier: identifier, isFavorite: status)
200             needSave = true
201         } else if favorite!.isFavorite != status {
202             // Otherwise, if the status is changing, then update the timestamp and record the new status.
203             favorite!.isFavorite = status
204             needSave = true
205         }
206
207         // Update the last modified date, if needed.
208         if favorite!.lastModified != lastModified {
209             favorite!.lastModified = lastModified
210             needSave = true
211         }
212
213         // If something changed, then store the updated favorite and kick off any related work, if needed.
```

**Decouple**

# Decouple

Determine functional segments and break them out

# Decouple

Determine functional segments and break them out

Scale your work across multiple apps

# Decouple

Determine functional segments and break them out

Scale your work across multiple apps

Improve efficiency with extensions



# Decouple

Determine functional segments and break them out

Scale your work across multiple apps

Improve efficiency with extensions

Share your efforts with the broader community

# Decouple

Determine functional segments and break them out

Scale your work across multiple apps

Improve efficiency with extensions

Share your efforts with the broader community

Documentation is critical

Organize

Track

Document

Test

Analyze

Evaluate

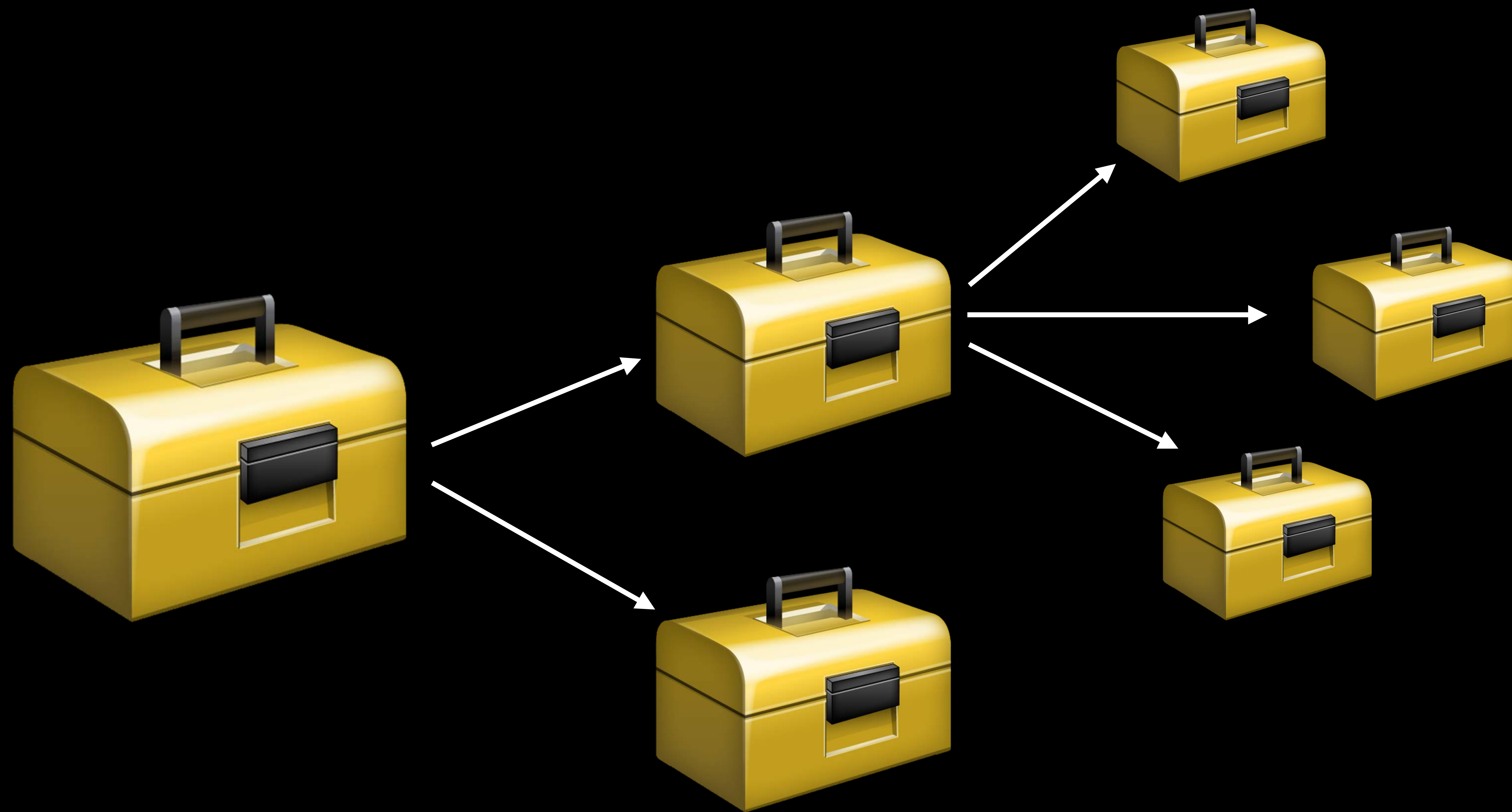
Decouple

Manage















Have a plan.

**Manage**

# Manage

Use community and open source projects responsibly

# Manage

Use community and open source projects responsibly

Understand dependencies thoroughly

# Manage

Use community and open source projects responsibly

Understand dependencies thoroughly

Ensure that privacy is respected

# Manage

Use community and open source projects responsibly

Understand dependencies thoroughly

Ensure that privacy is respected

Have a plan if a dependency goes away or is no longer maintained

**The Last 10%**



Organize

Track

Document

Test

Analyze

Evaluate

Decouple

Manage

Organize

**Track**

Document

Test

Analyze

Evaluate

Decouple

Manage

Organize

Track

**Document**

Test

Analyze

Evaluate

Decouple

Manage

Organize

Track

Document

**Test**

Analyze

Evaluate

Decouple

Manage

Organize

Track

Document

Test

**Analyze**

Evaluate

Decouple

Manage

Organize

Track

Document

Test

Analyze

**Evaluate**

Decouple

Manage

Organize

Track

Document

Test

Analyze

Evaluate

**Decouple**

Manage

Organize

Track

Document

Test

Analyze

Evaluate

Decouple

Manage



Organize

Track

Document

Test

Analyze

Evaluate

Decouple

Manage



# Simply beautiful app



This is the most elegant a  
care and attention to deta

# More Information

[developer.apple.com/wwdc19/239](https://developer.apple.com/wwdc19/239)

