

# What's New in tvOS

Session 209

Hans Kim, tvOS Engineering  
Marshall Huss, tvOS Engineering

# Agenda

# Agenda

SDK updates

# Agenda

SDK updates

View Controller enhancements

# SDK Updates







Fast initial download





Fast initial download

Quicker launch



Fast initial download

Quicker launch

High quality content



Fast initial download

Quicker launch

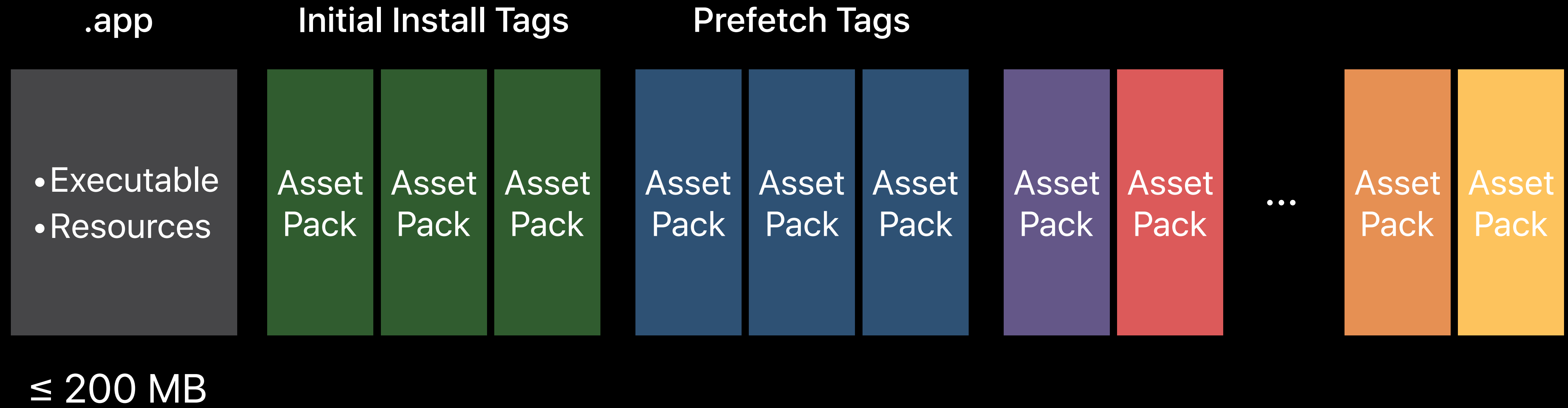
High quality content

Less local storage

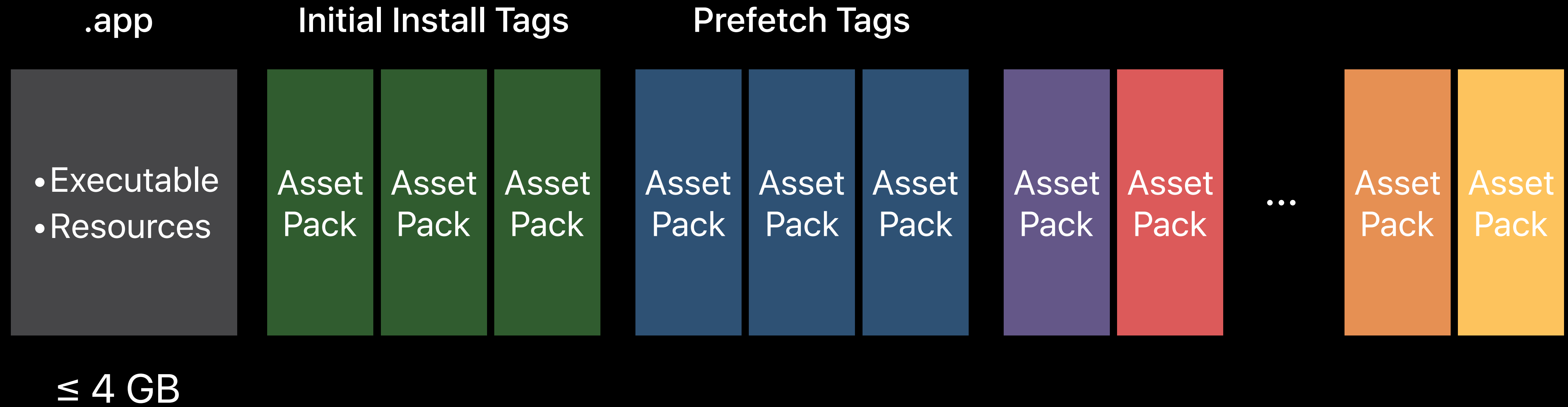
# On-Demand Resources



# On-Demand Resources



# On-Demand Resources



# On-Demand Resources

Take advantage of new limit

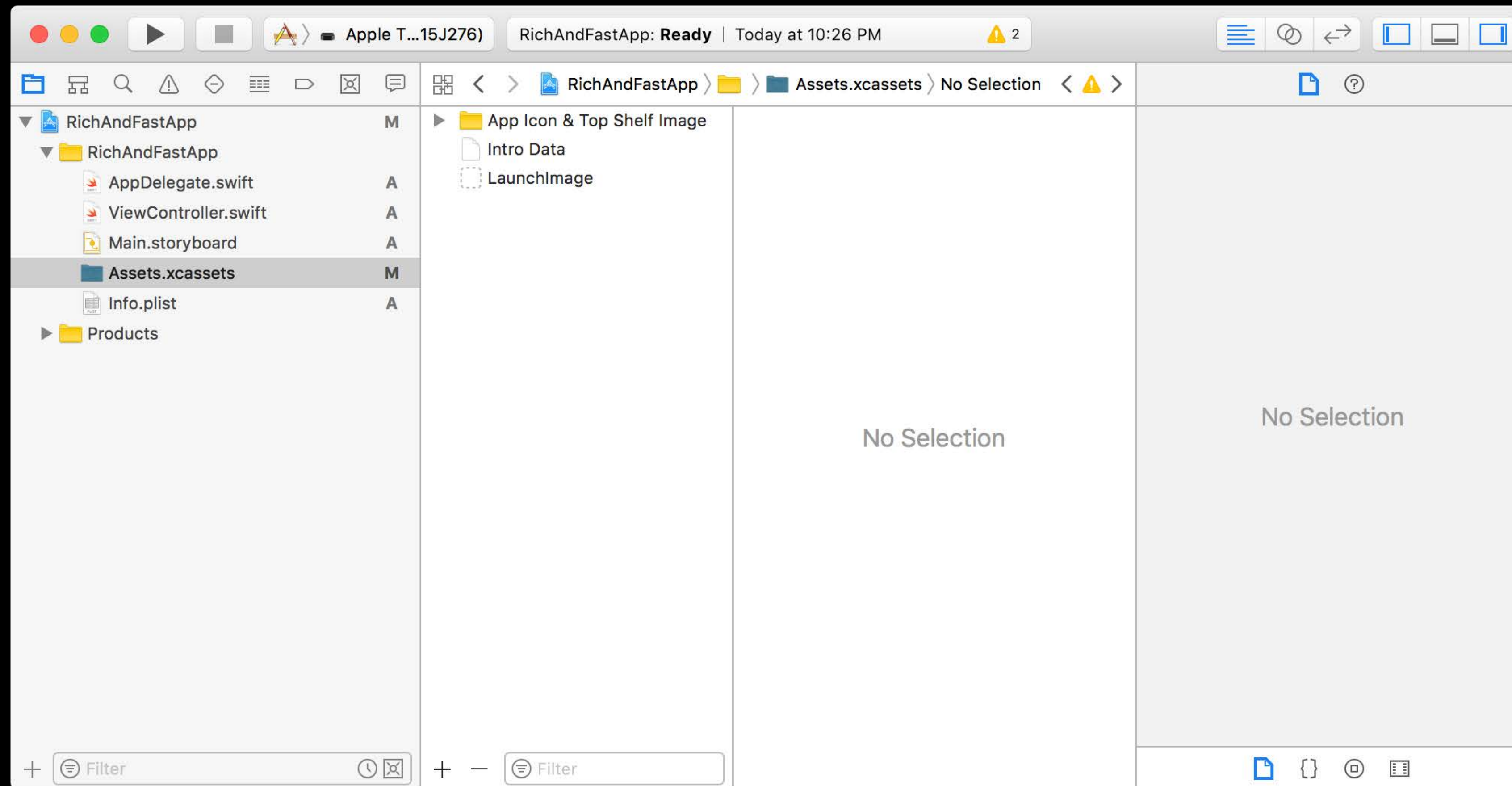


NEW

# On-Demand Resources

Take advantage of new limit

NEW

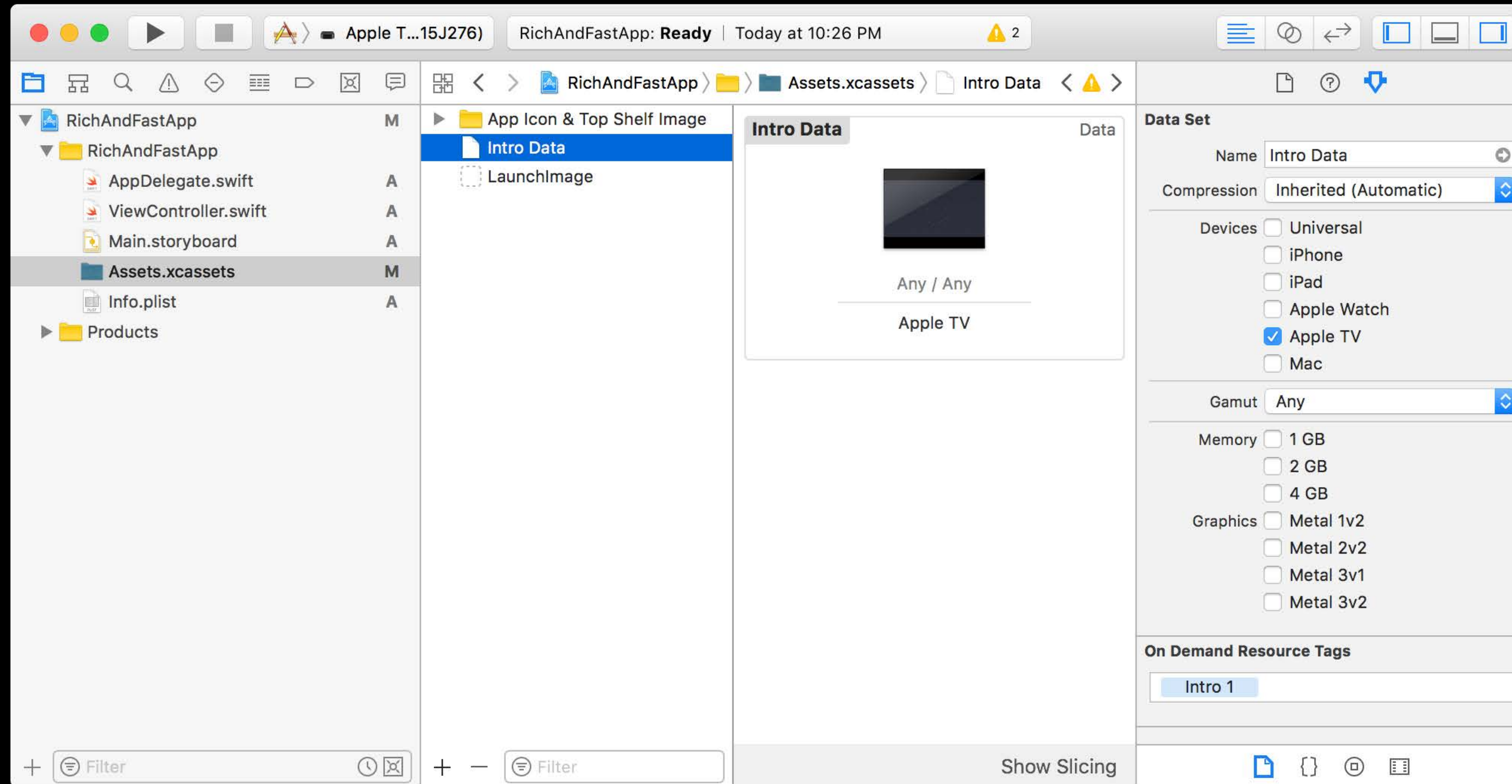




# On-Demand Resources

Take advantage of new limit

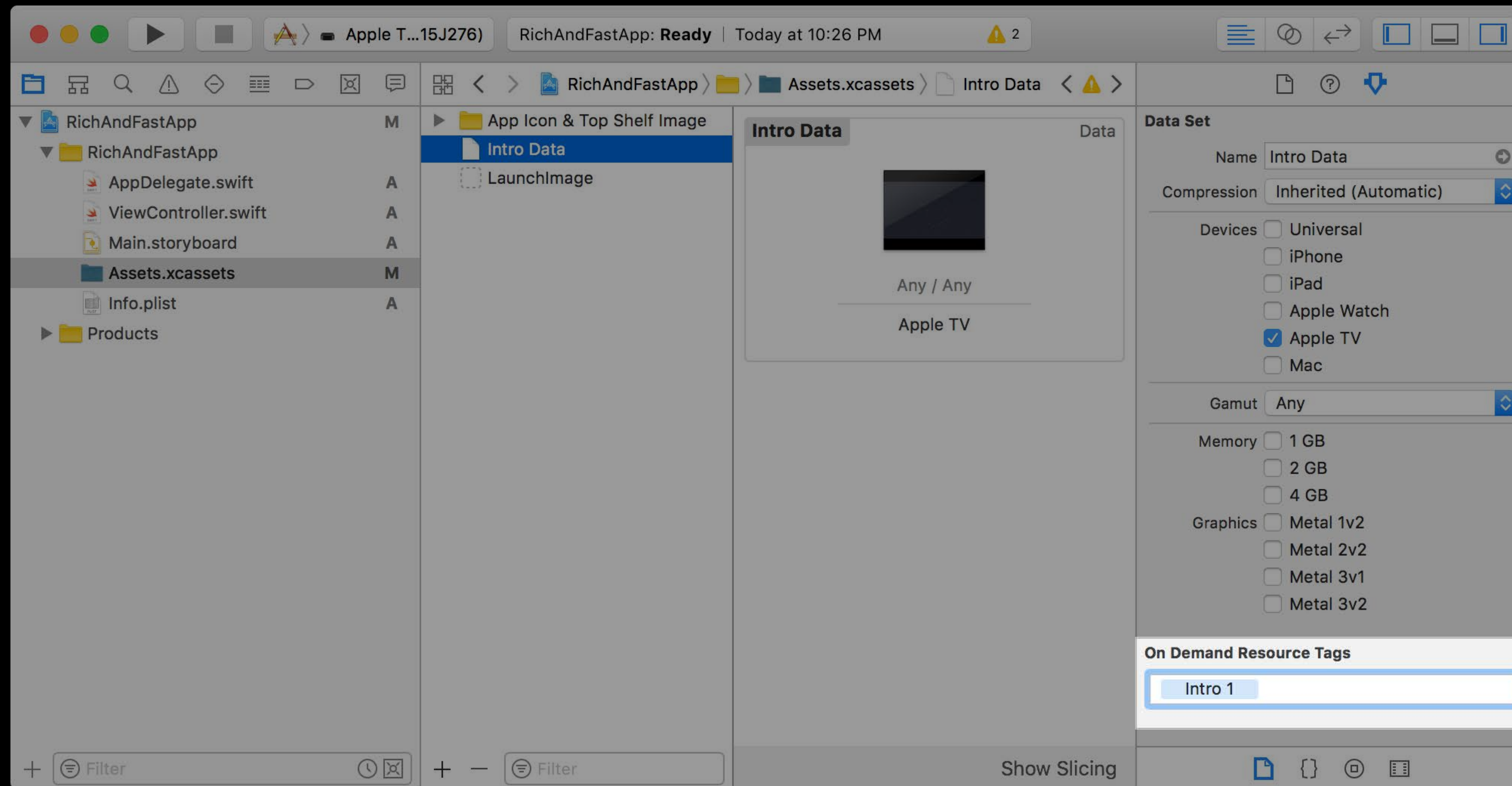
NEW



# On-Demand Resources

Take advantage of new limit

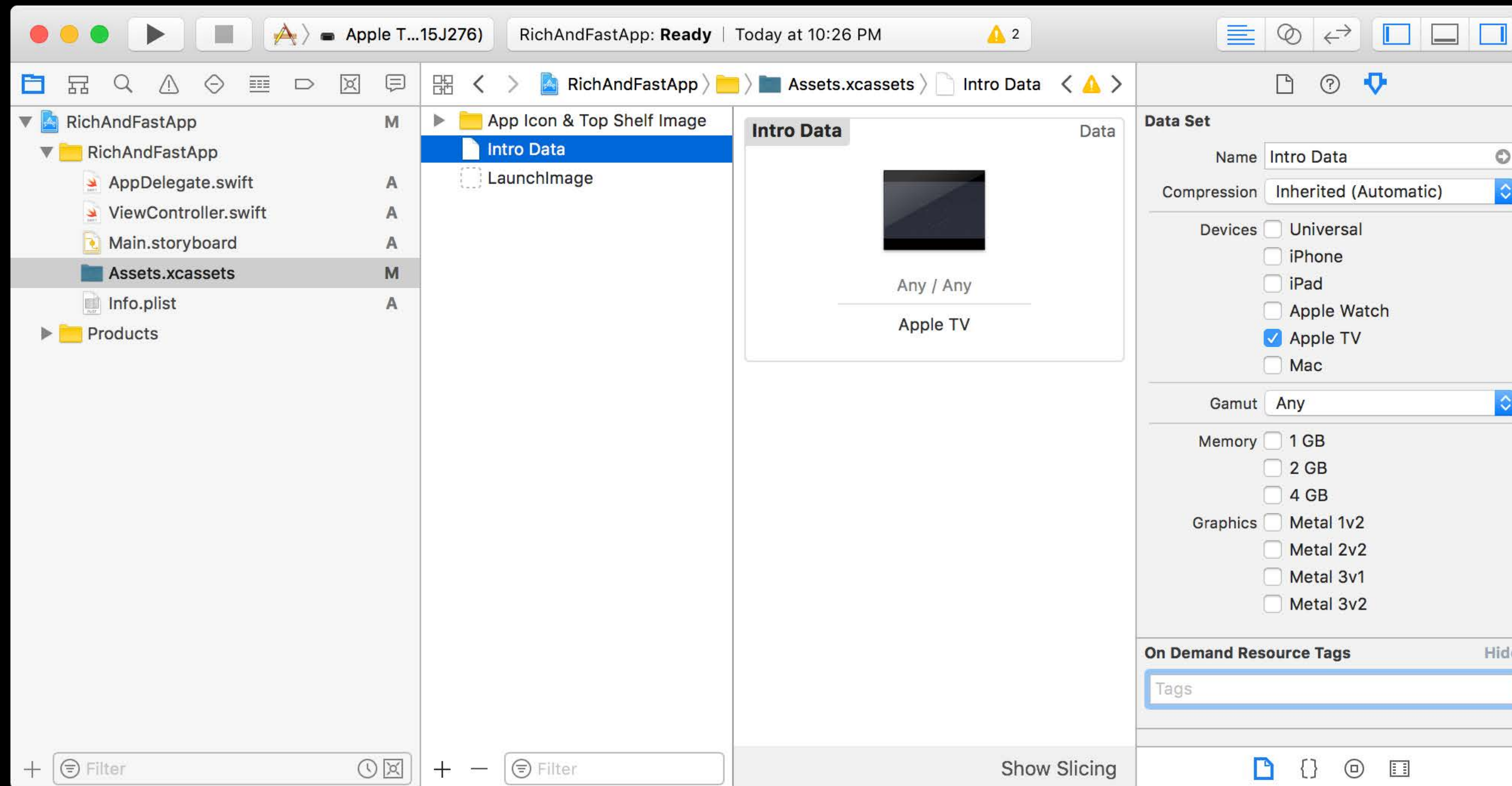
NEW



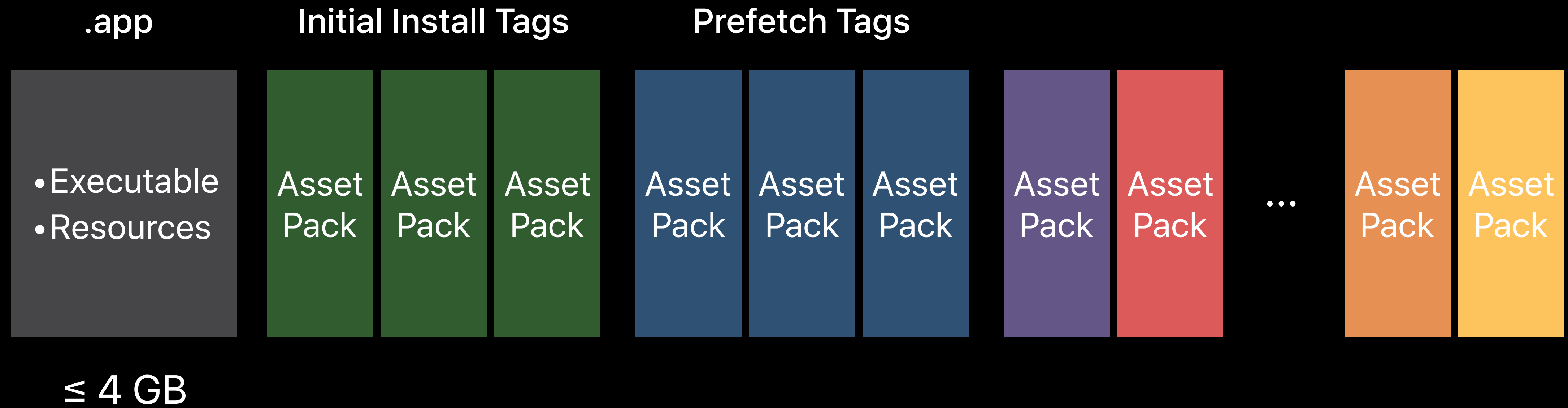
# On-Demand Resources

Take advantage of new limit

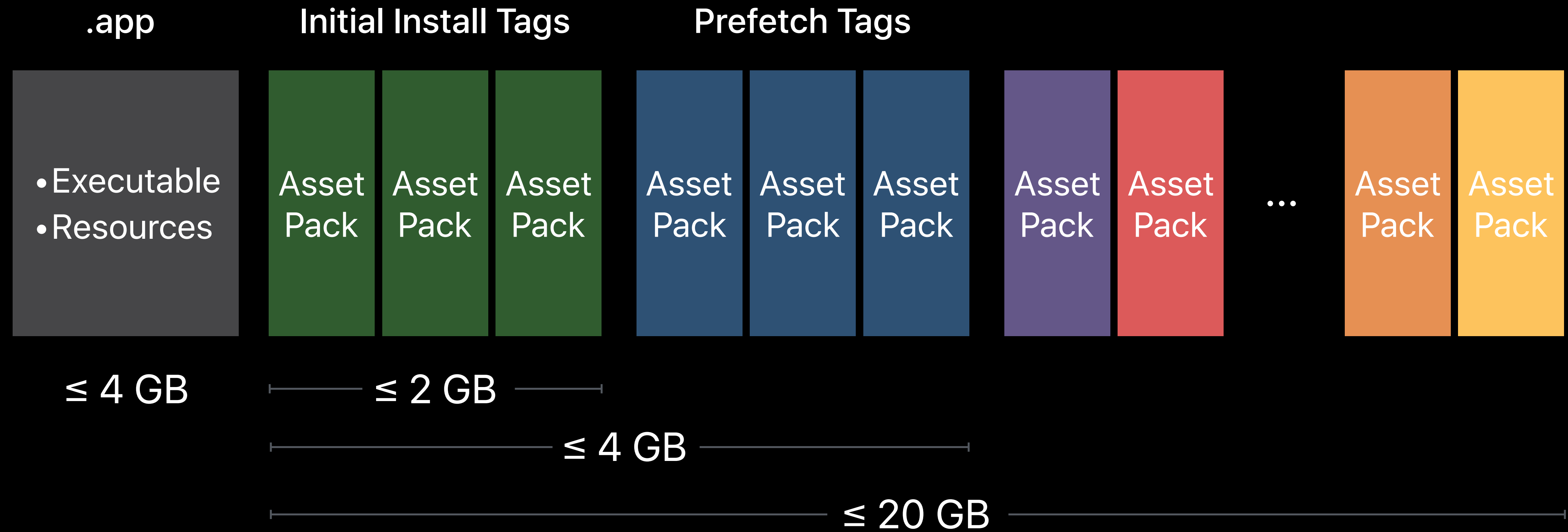
NEW



# On-Demand Resources



# On-Demand Resources



NEW

NEW

# 24 GB

App Bundle + Content

العربية

עברית



## AirPlay



With AirPlay, you can stream what's on your iOS device or computer to this Apple TV.

For more information, go to [support.apple.com/apptv](https://support.apple.com/apptv).

**AirPlay** On

Apple TV Name Apple TV

Conference Room Display Off >

Play Purchases from iCloud On

AirPlay Display Underscan Auto

### ACCESS CONTROL

**Security** None

**Access** Everyone

NEW

## AirPlay

تشغيل AirPlay

اسم Apple TV Apple TV

شاشة عرض غرفة الاجتماعات إيقاف <

تشغيل المشتريات من iCloud تشغيل

شاشة AirPlay داخل إطار الشاشة تلقائي

التحكم في الوصول

الأمن لا شيء

Access الجميع



مع AirPlay، يمكنك إجراء تدفق لما يظهر على جهاز iOS أو الكمبيوتر الخاص بك إلى هذا الـ Apple TV.

لمزيد من المعلومات، انتقل إلى:  
[support.apple.com/appletv](https://support.apple.com/appletv)

NEW

# AirPlay

**AirPlay** פעיל

שם ה-Apple TV Apple TV

תצוגת חדר ועידה < כבוי

נגן פריטים שנרכשו ב-iCloud פעיל

שוליים שחורים בצג AirPlay אוטומטי

בקרת גישה

אבטחה ללא

Access כולם



עם AirPlay, תוכלי להזרים תוכן ממכשיר ה-iOS או מהמחשב שלך ל-Apple TV זה.

לקבלת מידע נוסף, עבורי אל [support.apple.com/appletv](https://support.apple.com/appletv).

NEW

# AirPlay

- AirPlay
- פעיל
- שם ה-Apple TV
- שם ה-Apple TV
- פצות הדבר ועידה
- פעיל
- נגן פריטים שנרכשו ב-iCloud
- פעיל
- שוליים שחורים בצג AirPlay
- אוטומטי
- בקרת גישה
- לא
- כולם
- Access



עם AirPlay, תוכלו להזרים תוכן ממכשיר ה-iOS או מהמחשב שלך ל-Apple TV זה.  
לקבלת מידע נוסף, עבורו אל [support.apple.com/appletv](http://support.apple.com/appletv).

NEW



# RTL Support

## Tips

NEW

# RTL Support

Tips



NEW

Use Base Internationalization

# RTL Support

## Tips



NEW

### Use Base Internationalization

- Separates localization from layout



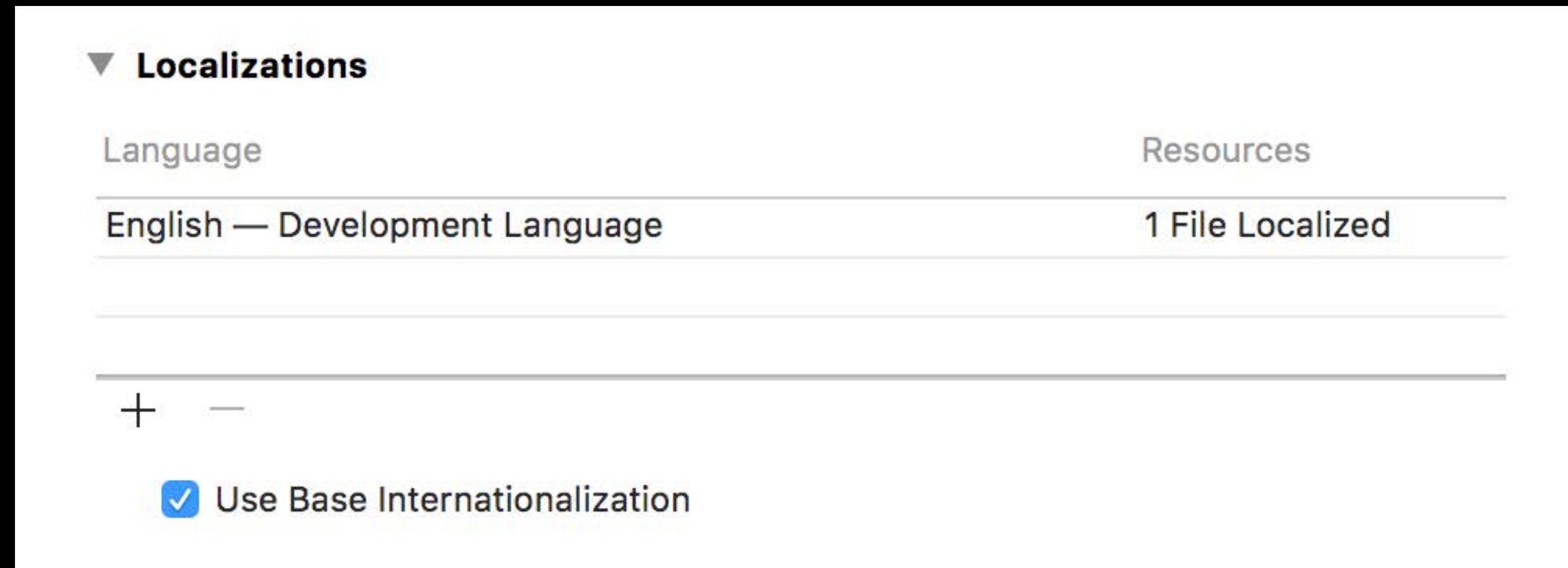
# RTL Support

## Tips

NEW

## Use Base Internationalization

- Separates localization from layout



# RTL Support

## Tips

NEW

# RTL Support

## Tips



Auto Layout

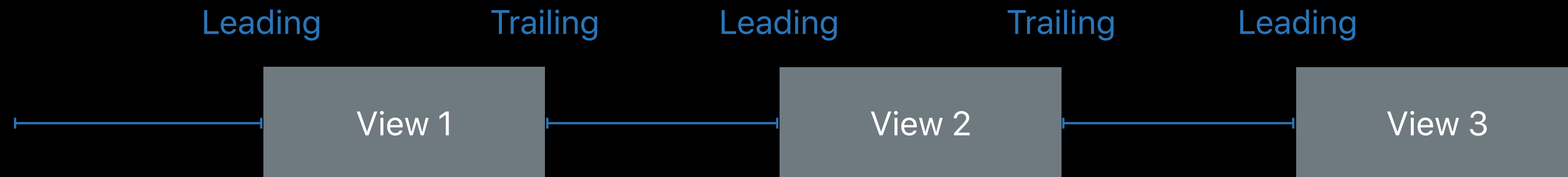
# RTL Support

## Tips

NEW

## Auto Layout

- Leading and trailing attributes



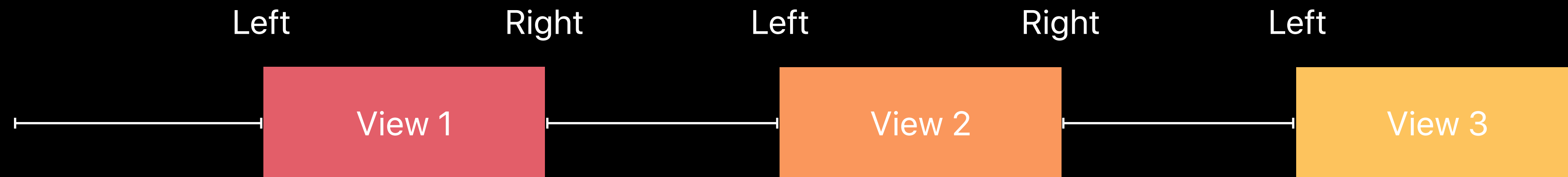
# RTL Support

## Tips

NEW

## Auto Layout

- Leading and trailing attributes



Left to Right Language

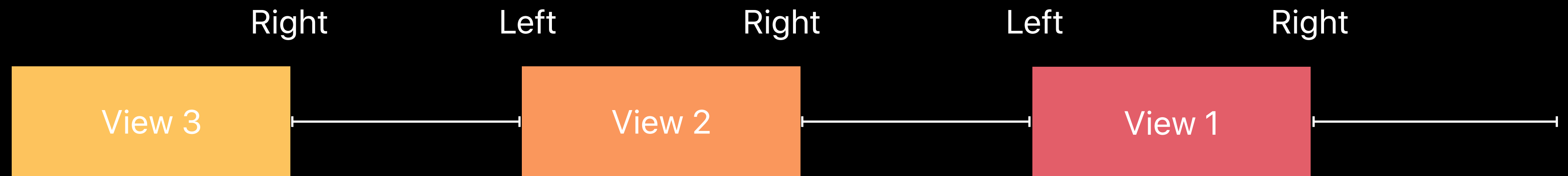
# RTL Support

## Tips

NEW

## Auto Layout

- Leading and trailing attributes



Right to Left Language

# RTL Support

## Tips

NEW

# RTL Support

Tips



NEW

Test during development



# RTL Support

Tips

NEW

Test during development

Application Language

System Language



# RTL Support

Tips

NEW

Test during development

Application Language

Arabic



# RTL Support

Tips

NEW

Test during development

Application Language

Hebrew



# RTL Support

Tips

NEW

Test during development

Application Language

Right to Left Pseudolanguage



# RTL Support

## Tips

NEW

## Test during development

Application Language

Right to Left Pseudolanguage



---

Internationalization Best Practices

WWDC 2016

---

What's New in International User Interfaces

WWDC 2016

---

# RTL Support

TVML

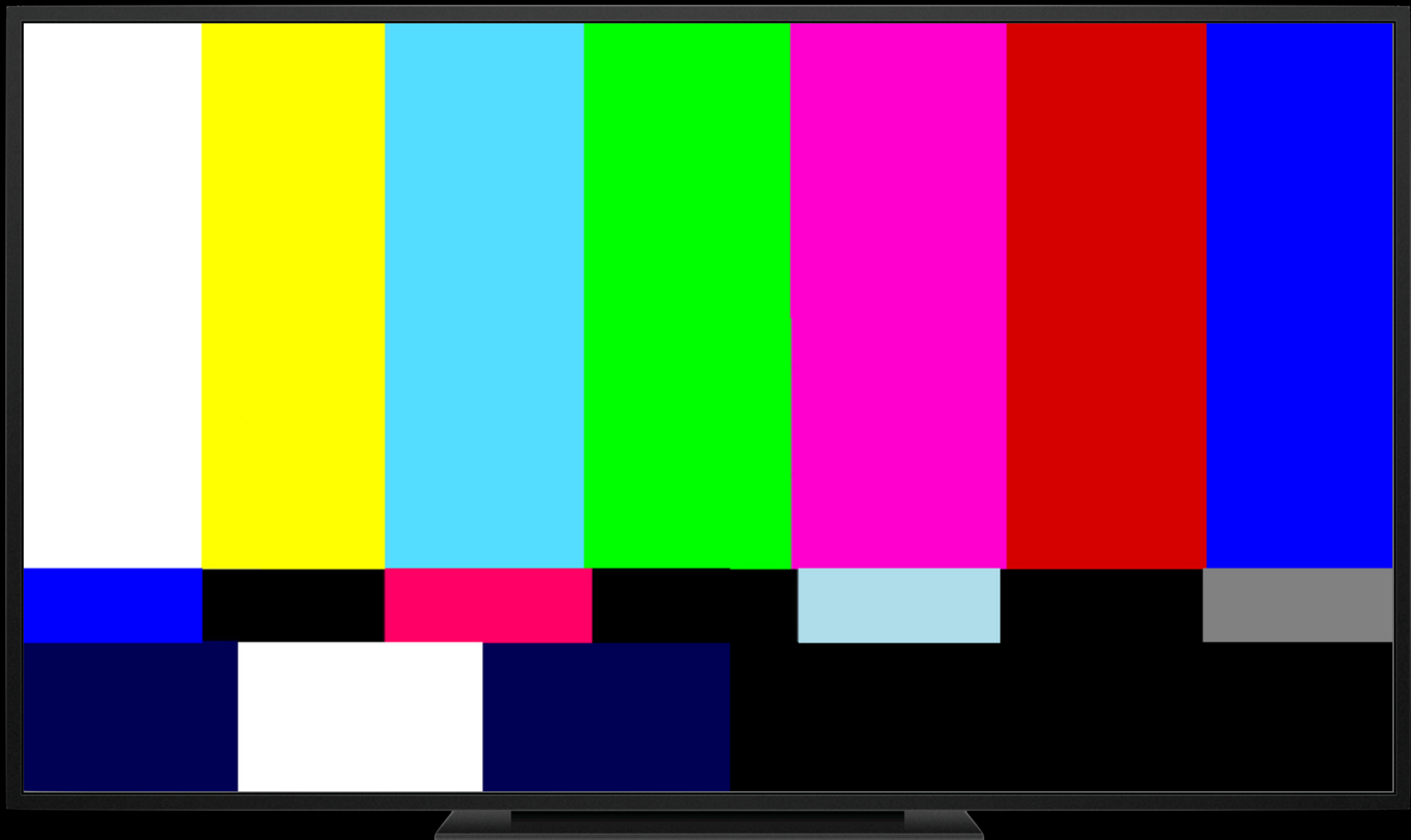
NEW



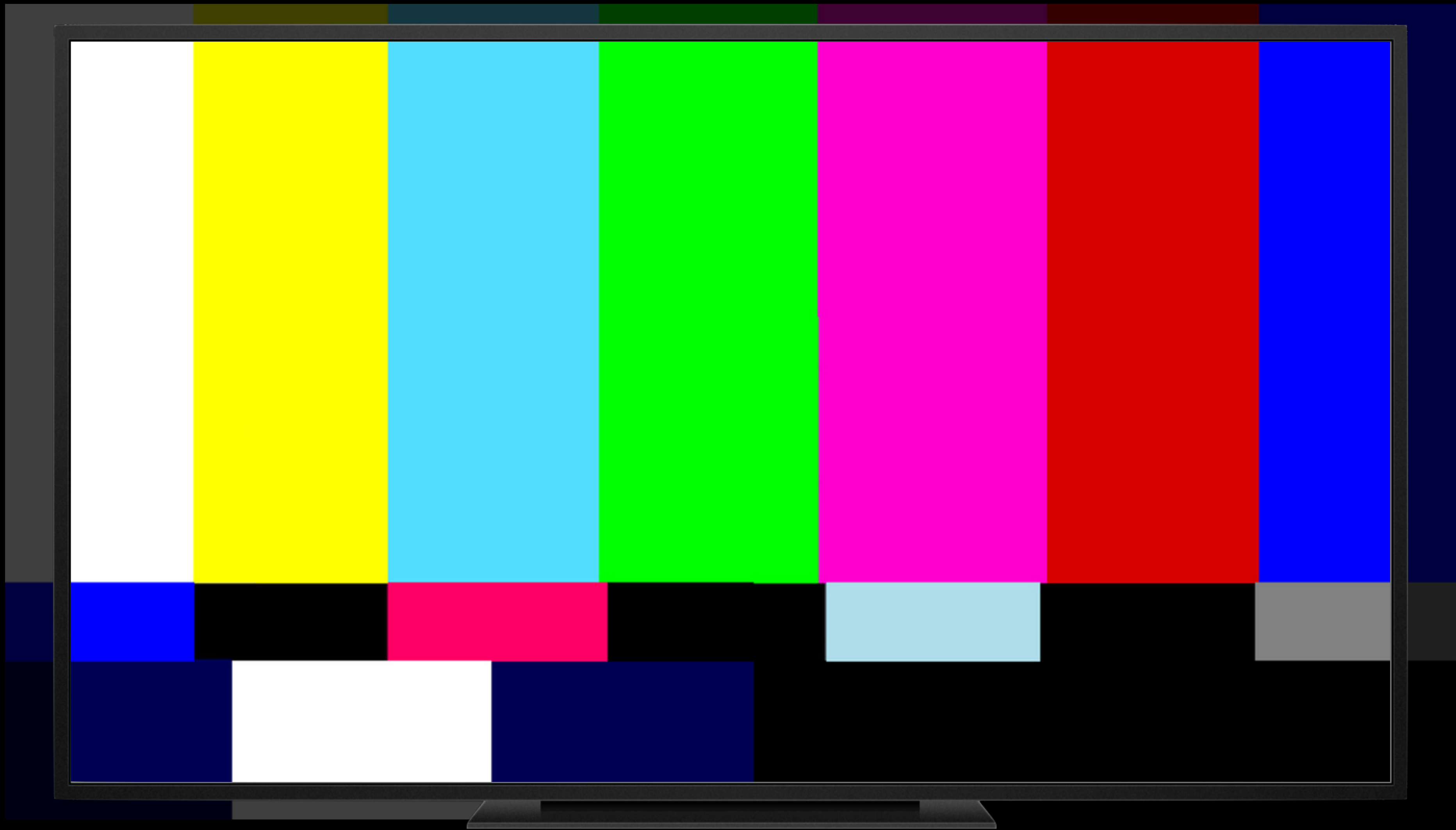
TVMLKit



Safe Area







# Overscan

tvOS 10

# Overscan

tvOS 10

Manually inset content

# Overscan

tvOS 10

## Manually inset content

```
UIScrollView.overscanCompensationInsets
```

# Safe Area

tvOS 11



NEW

# Safe Area

tvOS 11



Automatic

# Safe Area

tvOS 11

NEW

Automatic

```
UIView.safeAreaInsets
```

# Safe Area

tvOS 11



NEW

## Automatic

```
UIView.safeAreaInsets
```

```
UIView.safeAreaLayoutGuide
```



# Safe Area

tvOS 11

NEW

## Automatic

`UIView.safeAreaInsets`

`UIView.safeAreaLayoutGuide`

## Customization

# Safe Area

tvOS 11



NEW

## Automatic

`UIView.safeAreaInsets`

`UIView.safeAreaLayoutGuide`

## Customization

`UIViewController.additionalSafeAreaInsets`

# Safe Area

tvOS 11



NEW

## Automatic

```
UIView.safeAreaInsets
```

```
UIView.safeAreaLayoutGuide
```

## Customization

```
UIViewController.additionalSafeAreaInsets
```

Linking is opting-in

# Safe Area

tvOS 11

NEW

## Automatic

`UIView.safeAreaInsets`

`UIView.safeAreaLayoutGuide`

## Customization

`UIViewController.additionalSafeAreaInsets`

## Linking is opting-in

---

[Updating Your App for iOS 11](#)

Hall 3

Tuesday 4:10PM

---

# Safe Area

tvOS 11



NEW

# Safe Area

tvOS 11



Opting-out

# Safe Area

tvOS 11



NEW

## Opting-out

- Individual views, view controllers

# Safe Area

tvOS 11



NEW

## Opting-out

- Individual views, view controllers

```
view.insetsLayoutMarginsFromSafeArea = false
```

```
viewController.viewRespectsSystemMinimumLayoutMargins = false
```



# Safe Area

tvOS 11



NEW

## Opting-out

- Individual views, view controllers

```
view.insetsLayoutMarginsFromSafeArea = false
```

```
viewController.viewRespectsSystemMinimumLayoutMargins = false
```

# Safe Area

tvOS 11

NEW

## Opting-out

- Individual views, view controllers

```
view.insetsLayoutMarginsFromSafeArea = false
```

```
viewController.viewRespectsSystemMinimumLayoutMargins = false
```

# Safe Area

tvOS 11

NEW

## Opting-out

- Individual views, view controllers

```
view.insetsLayoutMarginsFromSafeArea = false
```

```
viewController.viewRespectsSystemMinimumLayoutMargins = false
```

- Scroll views

# Safe Area

tvOS 11

NEW

## Opting-out

- Individual views, view controllers

```
view.insetsLayoutMarginsFromSafeArea = false
```

```
viewController.viewRespectsSystemMinimumLayoutMargins = false
```

- Scroll views

```
scrollView.contentInsetAdjustmentBehavior = .never
```

Background Update

# UserNotification.framework

tvOS 10

# UserNotification.framework

tvOS 10

Push Notifications

# UserNotification.framework

tvOS 10

## Push Notifications

- System delivers only the last notification



# UserNotification.framework

tvOS 10

## Push Notifications

- System delivers only the last notification
- When the user launches the app

# UserNotification.framework

tvOS 11



NEW

# UserNotification.framework

tvOS 11



NEW

System wakes up your app in the background

# UserNotification.framework

tvOS 11



NEW

System wakes up your app in the background

Delivers every notification

# UserNotification.framework

tvOS 11



NEW

System wakes up your app in the background

Delivers every notification

Silent Notification

# Silent Notification

tvOS 11



NEW

# Silent Notification

tvOS 11



NEW

Wakes up your app in the background

# Silent Notification

tvOS 11



NEW

Wakes up your app in the background

To do work transparently for the user



# Silent Notification

tvOS 11



NEW

Wakes up your app in the background

To do work transparently for the user

- Download new content

# Silent Notification

tvOS 11



NEW

Wakes up your app in the background

To do work transparently for the user

- Download new content
- Refresh data

# Silent Notification

tvOS 11



NEW

Wakes up your app in the background

To do work transparently for the user

- Download new content
- Refresh data
- Set up in-app alert

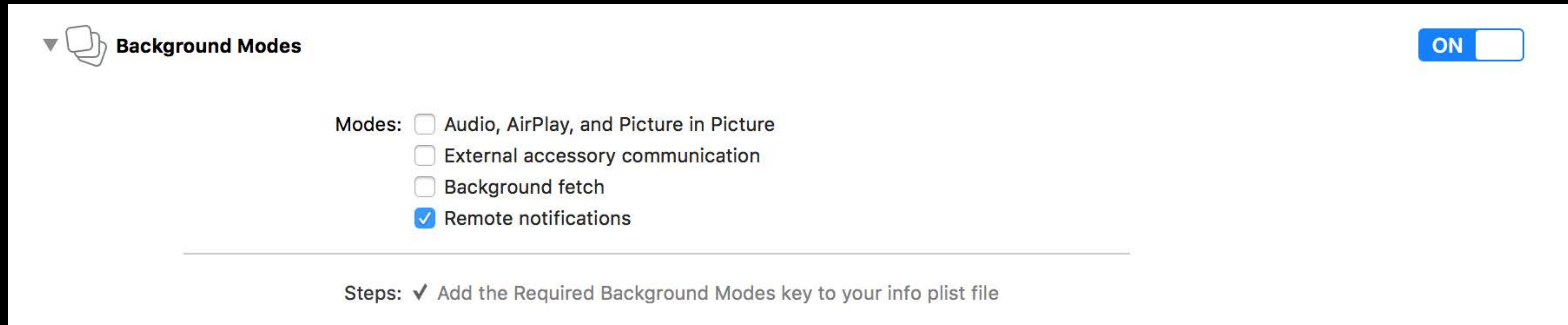
# Silent Notification Configuration



# Silent Notification Configuration

NEW

Add `Remote notifications` background mode





```
// Silent Notification
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    UNUserNotificationCenter.current().delegate = self // Become delegate
    application.registerForRemoteNotifications() // Register!

    return true
}
```

```
// Silent Notification
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    UNUserNotificationCenter.current().delegate = self // Become delegate

    application.registerForRemoteNotifications() // Register!

    return true
}
```



```
// Silent Notification
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    UNUserNotificationCenter.current().delegate = self // Become delegate

    application.registerForRemoteNotifications() // Register!

    return true
}

func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response:
UNNotification, withCompletionHandler completionHandler: @escaping () -> Void) {

    // download content
    // refresh data
    // set up in-app alert

}
```

```
// Silent Notification
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    UNUserNotificationCenter.current().delegate = self // Become delegate

    application.registerForRemoteNotifications() // Register!

    return true
}

func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response:
UNNotification, withCompletionHandler completionHandler: @escaping () -> Void) {

    // download content
    // refresh data
    // set up in-app alert

}
```

# Background Fetch

NEW

# Background Fetch

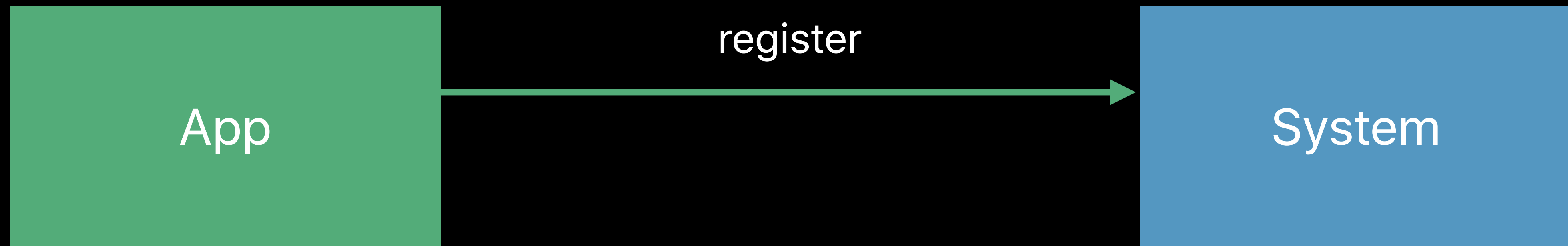
NEW



App

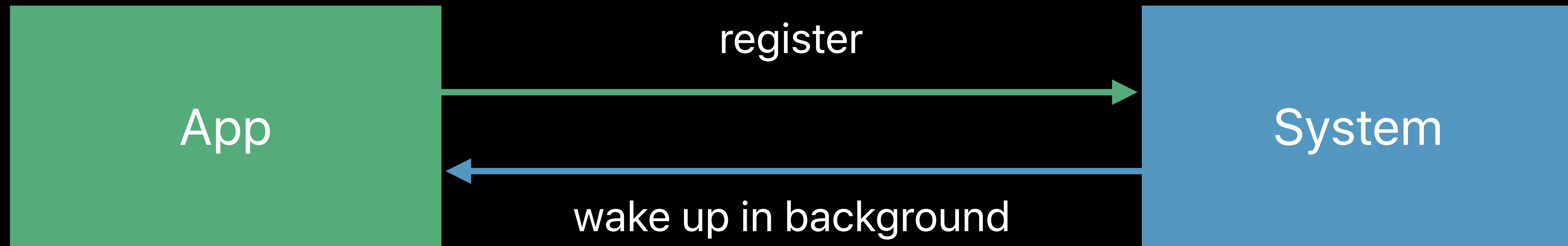
# Background Fetch

NEW



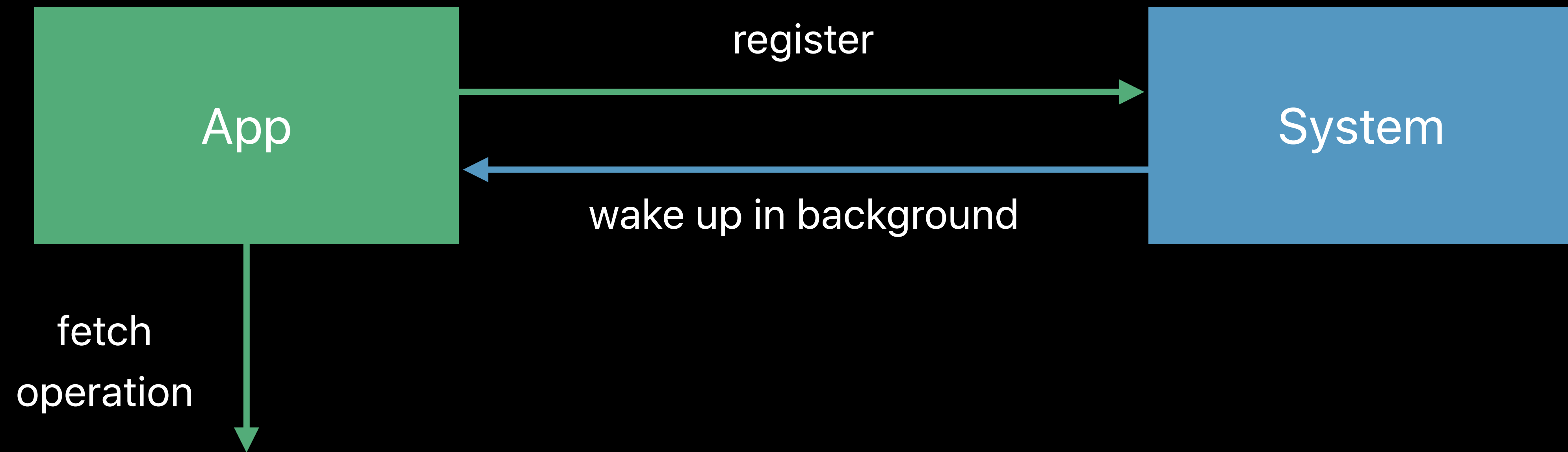
# Background Fetch

NEW



# Background Fetch

NEW



```
// Download content  
// Refresh Data  
// Set up in-app alert
```

# Background Fetch

## Configuration

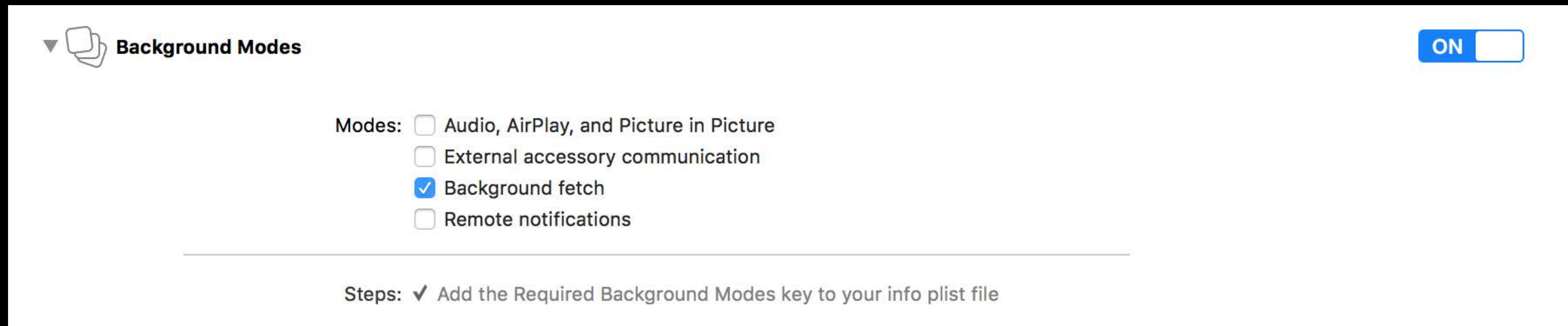




# Background Fetch Configuration

NEW

Add **Background fetch** background mode



```
// Background Fetch
```

```
// Configuration
```

```
// Background Fetch
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    // Register
    application.setMinimumBackgroundFetchInterval(UIApplicationBackgroundFetchIntervalMinimum)

    return true
}
```

```
// Background Fetch
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    // Register
    application.setMinimumBackgroundFetchInterval(UIApplicationBackgroundFetchIntervalMinimum)

    return true
}

func application(_ application: UIApplication, performFetchWithCompletionHandler
completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {

    // do background fetch

    completionHandler(.newData) // or .noData or .failed
}
```

```
// Background Fetch
// Configuration

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {

    // Register
    application.setMinimumBackgroundFetchInterval(UIApplicationBackgroundFetchIntervalMinimum)

    return true
}

func application(_ application: UIApplication, performFetchWithCompletionHandler
completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {

    // do background fetch

    completionHandler(.newData) // or .noData or .failed
}
```

# Background Work

2 new ways



NEW

# Background Work

2 new ways



---

Data availability

---

---

# Background Work

2 new ways



---

Data availability

Silent Notification

---

---



# Background Work

2 new ways



---

Data availability

Silent Notification

---

System availability

---

# Background Work

2 new ways



NEW

---

Data availability

Silent Notification

---

System availability

Background Fetch

---

# Background Work

2 new ways



NEW

---

Data availability

Silent Notification

---

System availability

Background Fetch

---

- Keep your app up-to-date before entering foreground

Scrolling

NEW

## Region



Afghanistan

Åland Islands

Albania

Algeria

American Samoa

Andorra

Angola

Anguilla

Antarctica

Antigua & Barbuda

Argentina

NEW

## Region



Afghanistan

Åland Islands

Albania

Algeria

American Samoa

Andorra

Angola

Anguilla

Antarctica

Antigua & Barbuda

Argentina

NEW



NEW





NEW



NEW



# Accelerated Scrolling

NEW

# Accelerated Scrolling

NEW

Comes free in UIScrollView

# Accelerated Scrolling



NEW

Comes free in UIScrollView

- UITableView

# Accelerated Scrolling



NEW

Comes free in UIScrollView

- UITableView
- UICollectionView

# Accelerated Scrolling



NEW

Comes free in UIScrollView

- UITableView
- UICollectionView

Behavior always on

# Accelerated Scrolling

## Customization



NEW



# Accelerated Scrolling

Customization



NEW

Index bar visibility

```
scrollView.indexDisplayMode = .automatic
```

# Accelerated Scrolling

## Customization

NEW

### Index bar visibility

```
scrollView.indexDisplayMode = .automatic // or .hidden
```

# Accelerated Scrolling

## Customization

NEW

### Index bar visibility

```
scrollView.indexDisplayMode = .automatic // or .hidden
```

### Scroll indicator style

```
scrollView.indicatorStyle = .default
```

# Accelerated Scrolling

## Customization

NEW

### Index bar visibility

```
scrollView.indexDisplayMode = .automatic // or .hidden
```

### Scroll indicator style

```
scrollView.indicatorStyle = .default // or .black/.white
```

# Accelerated Scrolling

## Customization

NEW

# Accelerated Scrolling

Customization



Table View index titles

# Accelerated Scrolling

## Customization



NEW

### Table View index titles

```
optional public func sectionIndexTitles(for tableView: UITableView) -> [String]?
```

# Accelerated Scrolling

## Customization



NEW

## Table View index titles

```
optional public func sectionIndexTitles(for tableView: UITableView) -> [String]?  
optional public func tableView(_ tableView: UITableView, sectionForSectionIndexTitle title:  
    String, at index: Int) -> Int
```



# Accelerated Scrolling

## Customization



NEW

### Table View index titles

```
optional public func sectionIndexTitles(for tableView: UITableView) -> [String]?  
optional public func tableView(_ tableView: UITableView, sectionForSectionIndexTitle title:  
    String, at index: Int) -> Int
```

### Collection View index titles

# Accelerated Scrolling

## Customization



NEW

### Table View index titles

```
optional public func sectionIndexTitles(for tableView: UITableView) -> [String]?  
optional public func tableView(_ tableView: UITableView, sectionForSectionIndexTitle title:  
    String, at index: Int) -> Int
```

### Collection View index titles

```
optional public func indexTitles(for collectionView: UICollectionView) -> [String]?
```

# Accelerated Scrolling

## Customization



NEW

### Table View index titles

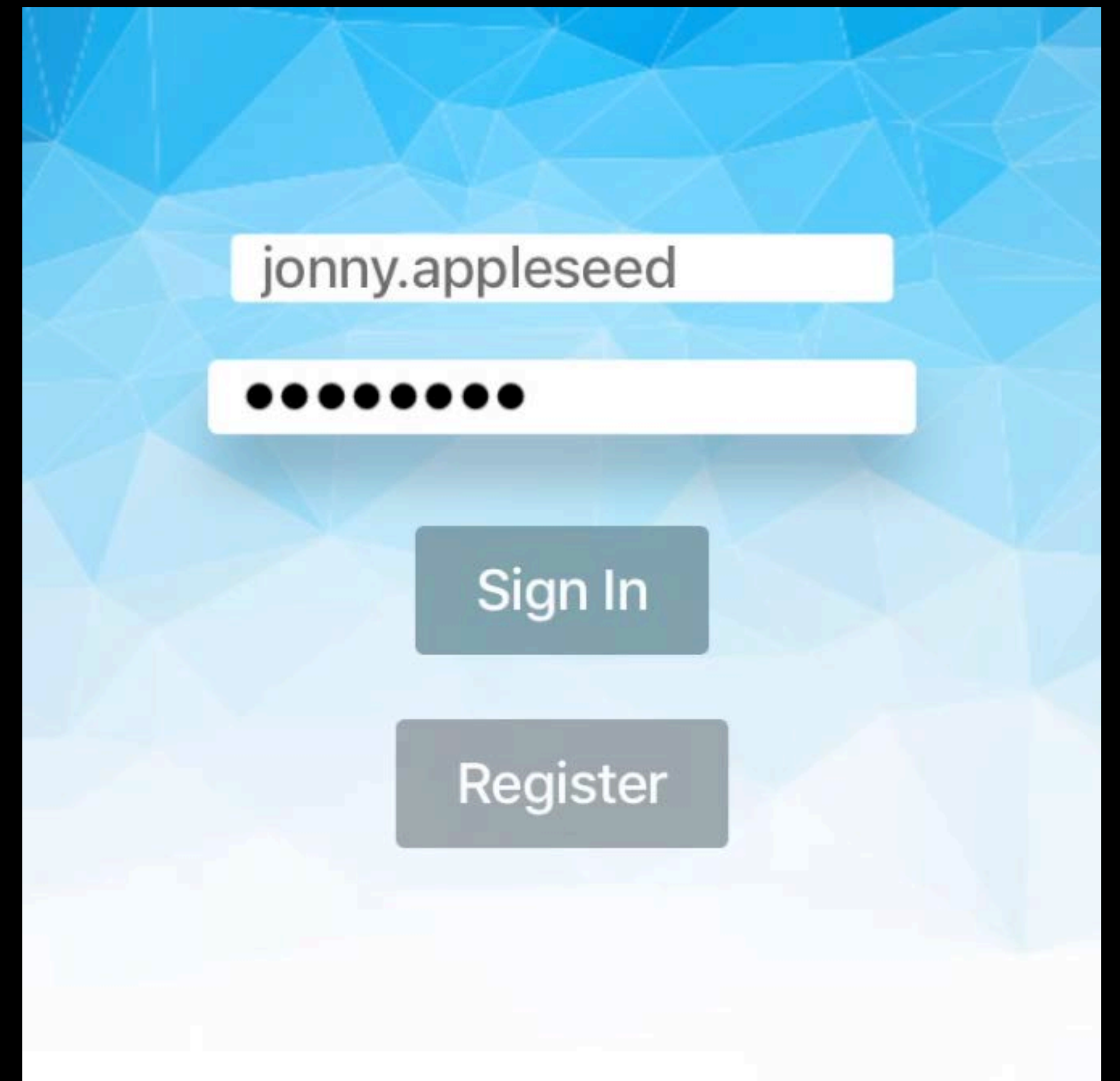
```
optional public func sectionIndexTitles(for tableView: UITableView) -> [String]?  
optional public func tableView(_ tableView: UITableView, sectionForSectionIndexTitle title:  
    String, at index: Int) -> Int
```

### Collection View index titles

```
optional public func indexTitles(for collectionView: UICollectionView) -> [String]?  
optional public func collectionView(_ collectionView: UICollectionView, indexPathForIndexTitle  
    title: String, at index: Int) -> IndexPath
```

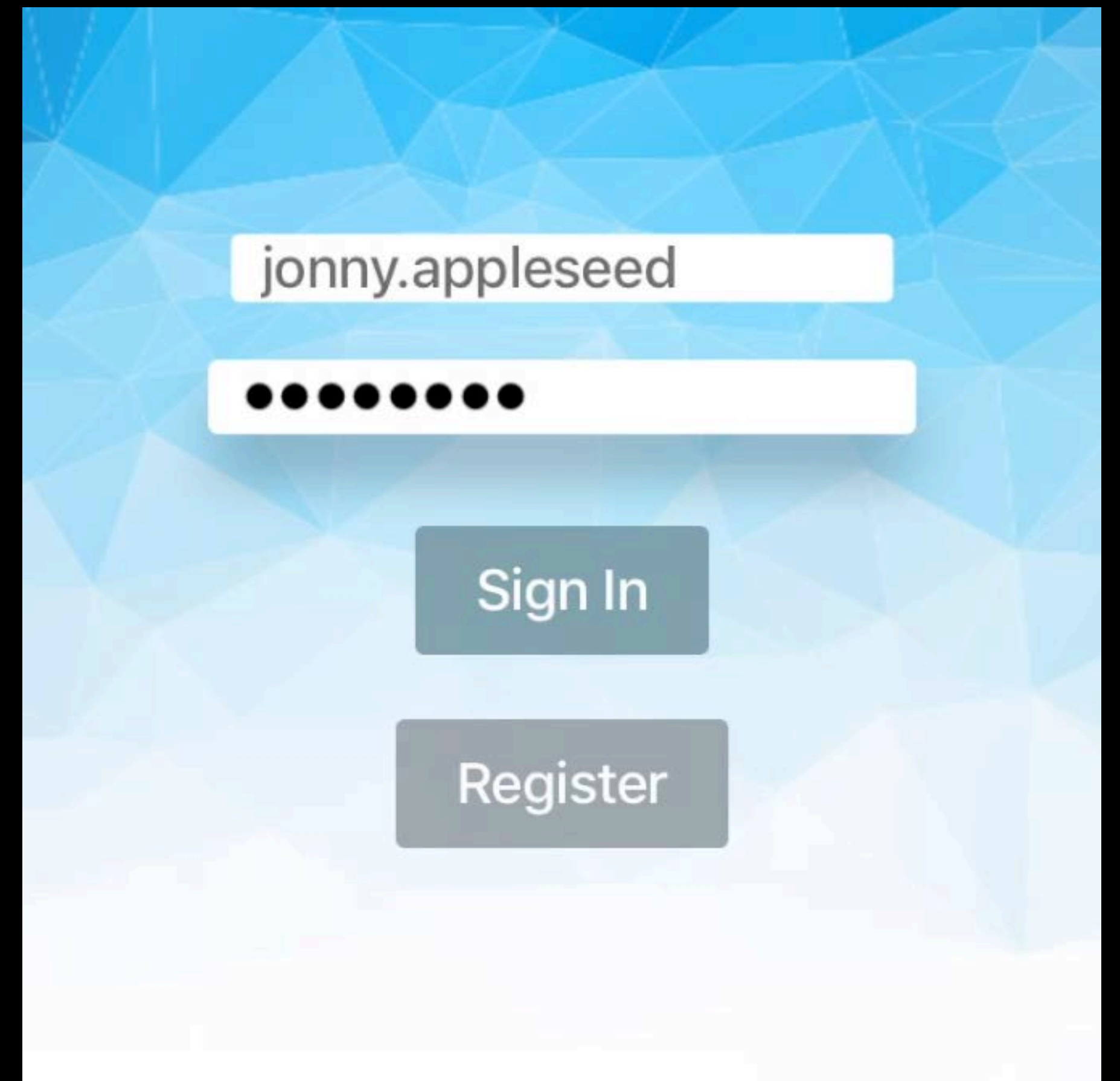
UIButtonTypePlain

# UIButtonTypeSystem



# UIButtonTypeSystem

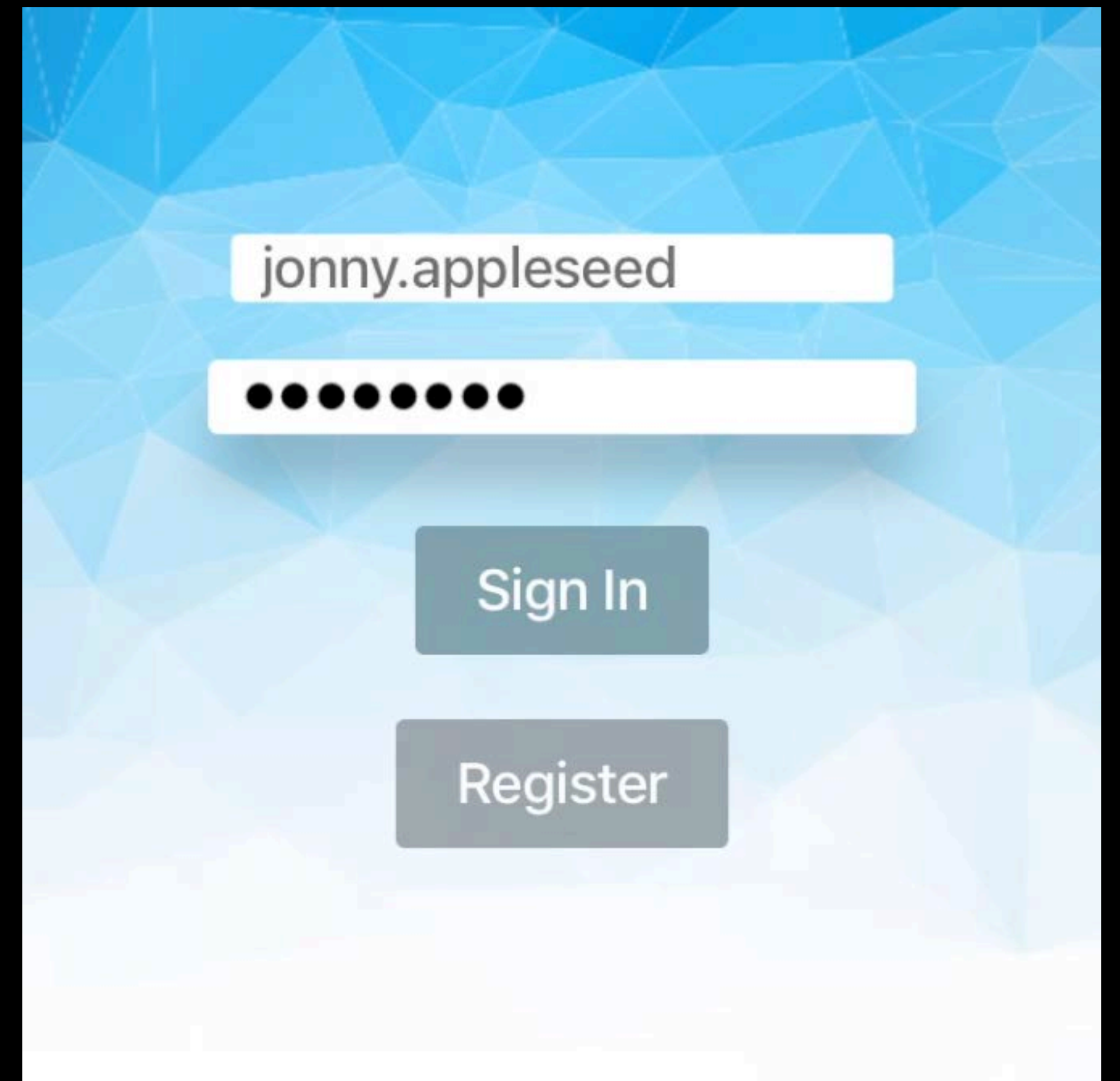
Interactive focus effect



# UIButtonTypeSystem

Interactive focus effect

Background blur

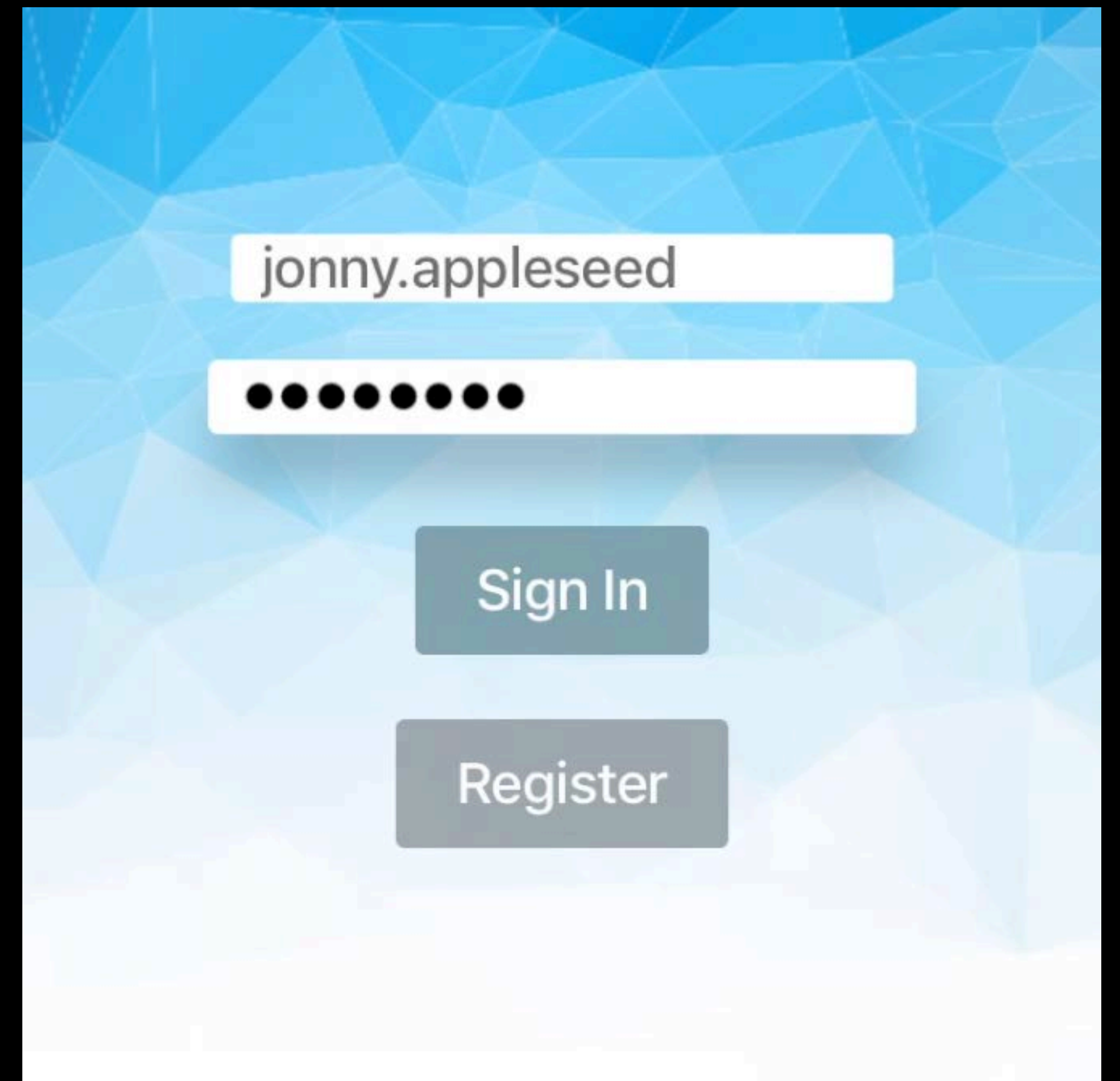


# UIButtonTypeSystem

Interactive focus effect

Background blur

- Affordance



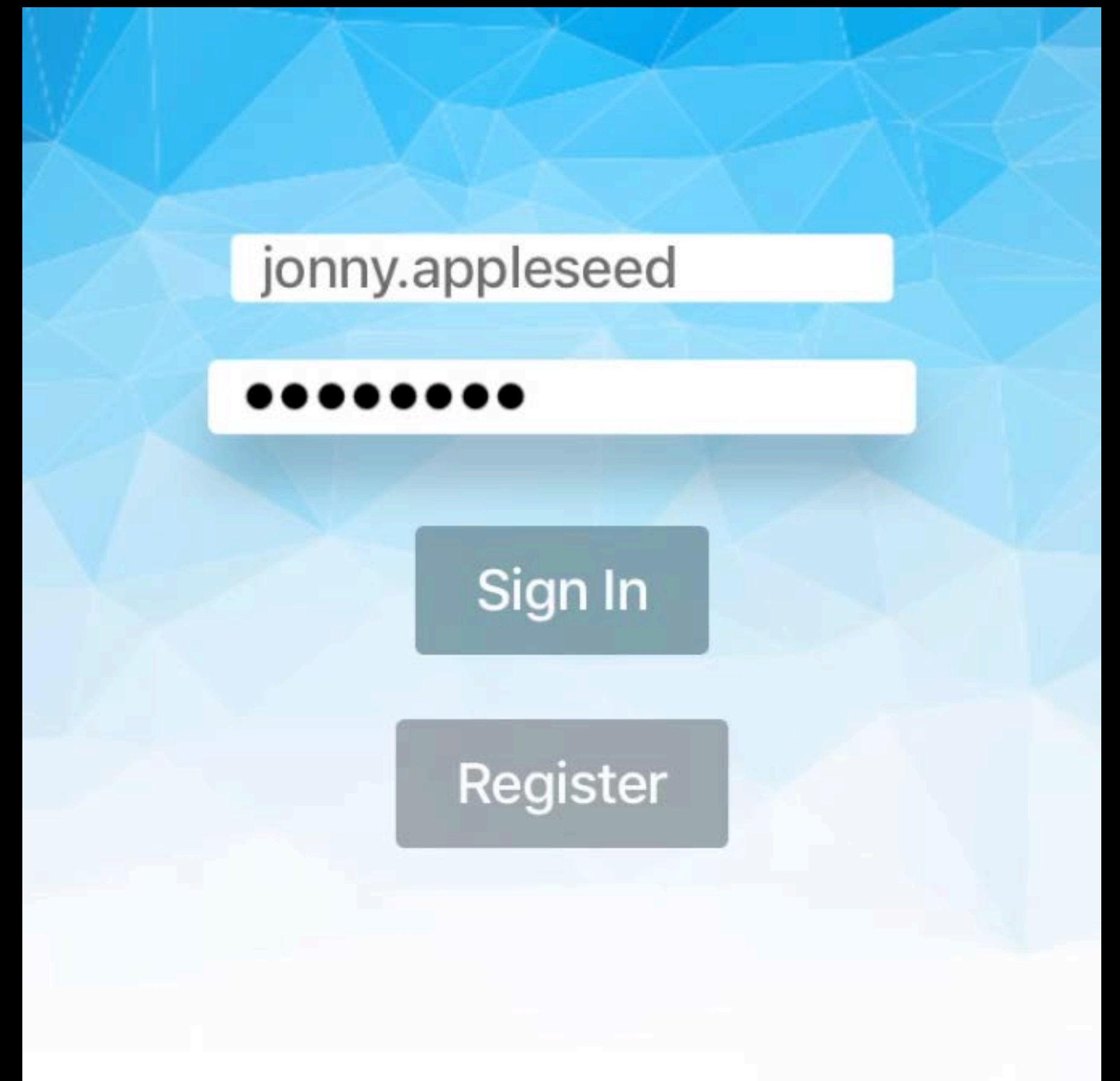


# UIButtonTypeSystem

Interactive focus effect

Background blur

- Affordance
- Title legibility



# UIButtonTypePlain

NEW

jonny.appleseed

●●●●●●●●

Sign In

Register

# UIButtonTypePlain

NEW

Interactive focus effect



# UIButtonTypePlain

NEW

Interactive focus effect

No background blur



# UIButtonTypePlain

NEW

Interactive focus effect

No background blur

- Full customization



# UIButtonTypePlain

Usage



NEW

# UIButtonTypePlain

## Usage

NEW

```
let button = UIButton(type: .plain)
```

# UIButtonTypePlain

## Usage



NEW

```
let button = UIButton(type: .plain)
button.backgroundColor = themeColor
...
```



UIImageView





# Image Overlay

NEW

# Image Overlay



NEW

```
UIImageView.overlayContentView
```

# Image Overlay



NEW

`UIImageView.overlayContentView`

- Add your content

# Image Overlay



NEW

`UIImageView.overlayContentView`

- Add your content
- Clips to imageView's bounds

# Image Overlay

NEW

`UIImageView.overlayContentView`

- Add your content
- Clips to imageView's bounds

```
imageView.overlayContentView.clipsToBounds = false
```



# Alpha Channel

# Alpha Channel

---

Layered Image

---

---

# Alpha Channel

---

Layered Image

Opaque background

---

---

# Alpha Channel

---

Layered Image

Opaque background

---

Flat Image

---

# Alpha Channel

---

Layered Image

Opaque background

---

Flat Image

Opaque background

---

# Alpha Channel

NEW

---

Layered Image

Opaque background

---

Flat Image

Any background

---

# Alpha Channel

NEW

---

Layered Image

Opaque background

---

Flat Image

Any background

---

Arbitrary shape alpha channel

# Alpha Channel

NEW

---

Layered Image

Opaque background

---

Flat Image

Any background

---

Arbitrary shape alpha channel

- Non-rectangular shadows



# *Demo*

Paul Schneider, tvOS Engineering

Audio Toolbox FairPlay Streaming Security Framework CoreText  
Foundation System Configuration Media Player Core Animation  
Core Audio SceneKit Image IO Core Data AVFoundation  
Quartz Core Speech Framework  
UIKit  
Open AL OpenGL ES  
SpriteKit Core Foundation  
Media Accessibility System Configuration Framework MapKit  
Multipeer Connectivity CloudKit AVKit Core Location  
Store Kit Javascript Core Mobile Core Services HomeKit

# ios



Audio Toolbox FairPlay Streaming Security Framework CoreText

Foundation System Configuration Media Player Core Animation

Core Audio SceneKit Image IO Core Data AVFoundation

Quartz Core Speech Framework

UIKit  
Open AL OpenGL ES

SpriteKit Core Foundation

Media Accessibility System Configuration Framework MapKit

Multipeer Connectivity CloudKit AVKit Core Location

Store Kit Javascript Core Mobile Core Services HomeKit

# tvOS

Audio Toolbox FairPlay Streaming Security Framework CoreText

Foundation System Configuration Media Player Core Animation

Core Audio SceneKit Image IO Core Data AVFoundation

Quartz Core Speech Framework

UIKit  
Open AL OpenGL ES

SpriteKit Core Foundation

Media Accessibility System Configuration Framework MapKit

Multipeer Connectivity CloudKit **AVKit** Core Location

Store Kit Javascript Core Mobile Core Services HomeKit

# tvOS

# What's New in AVKit

Dan Wright, Media Frameworks Engineering

# AVPlayerViewController

Standard interactive video playback

Scrubbing, skipping, scanning

All input methods

Interstitials, content proposals, and more

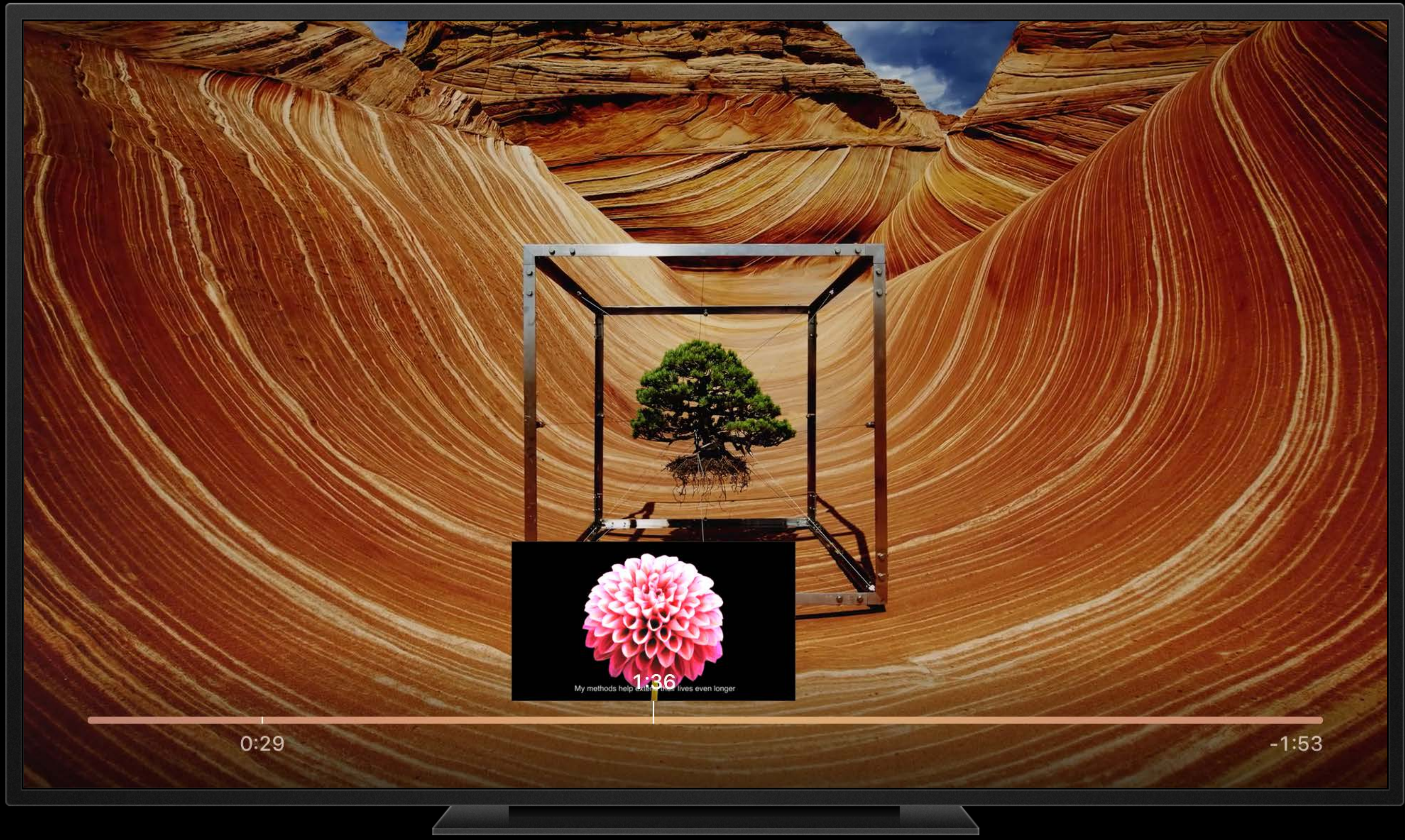
# AVPlayerViewController

Standard interactive video playback

Scrubbing, skipping, scanning

All input methods

Interstitials, content proposals, and more







Episode 2 - More Scenic Beauties



Play Next

Return To Video

In this episode we will explore beautiful scenery from around the world. We will see breathtaking views of mountains, fields, sunsets, and cities from some spectacular places on this planet.

MENU



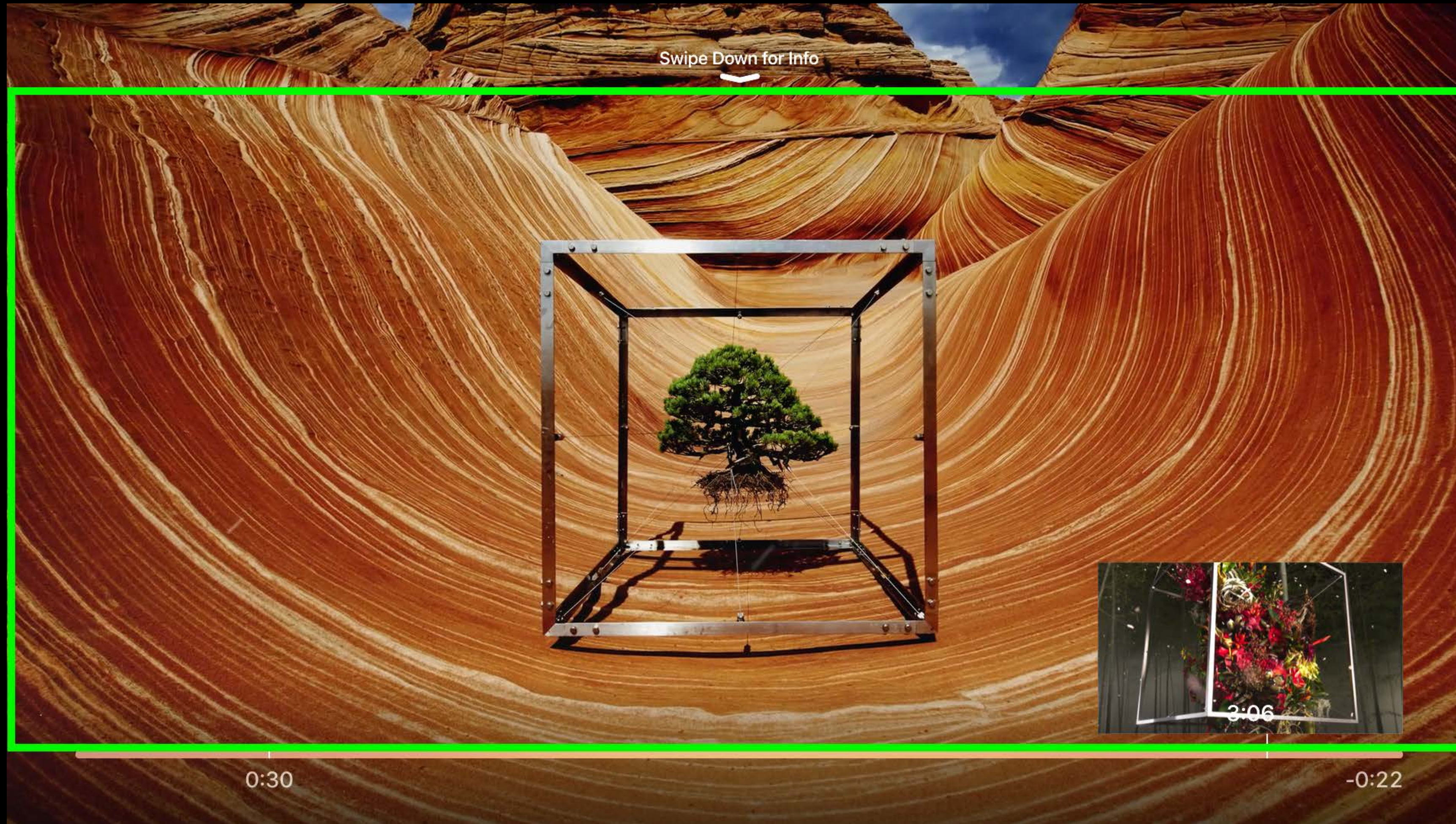
# Unobscured Content Guide

NEW



# Unobscured Content Guide

NEW



# Transport Bar Animation Coordinator

NEW

Showing a logo with the transport bar

# Transport Bar Animation Coordinator



NEW

## Showing a logo with the transport bar

```
func playerViewController(_ playerViewController: AVPlayerViewController,
willTransitionToVisibilityOfTransportBar visible: Bool, with coordinator:
AVPlayerViewControllerAnimationCoordinator) {
    coordinator.addCoordinatedAnimations({
        self.logoImageView.alpha = visible ? 1.0 : 0.0
    }, completion: nil)
}
```

# Controlling Dismissal

NEW

Delegate methods

# Controlling Dismissal



NEW

## Delegate methods

```
public protocol AVPlayerViewControllerDelegate : NSObjectProtocol {  
  
    optional public func playerViewControllerShouldDismiss(_ playerViewController:  
AVPlayerViewController) -> Bool  
  
    optional public func playerViewControllerWillBeginDismissalTransition(_  
playerViewController: AVPlayerViewController)  
  
    optional public func playerViewControllerDidEndDismissalTransition(_ playerViewController:  
AVPlayerViewController)  
}
```

Info Audio

SOUND

- ✓ Full Dynamic Range
- Reduce Loud Sounds

SPEAKER

- ✓  Apple TV

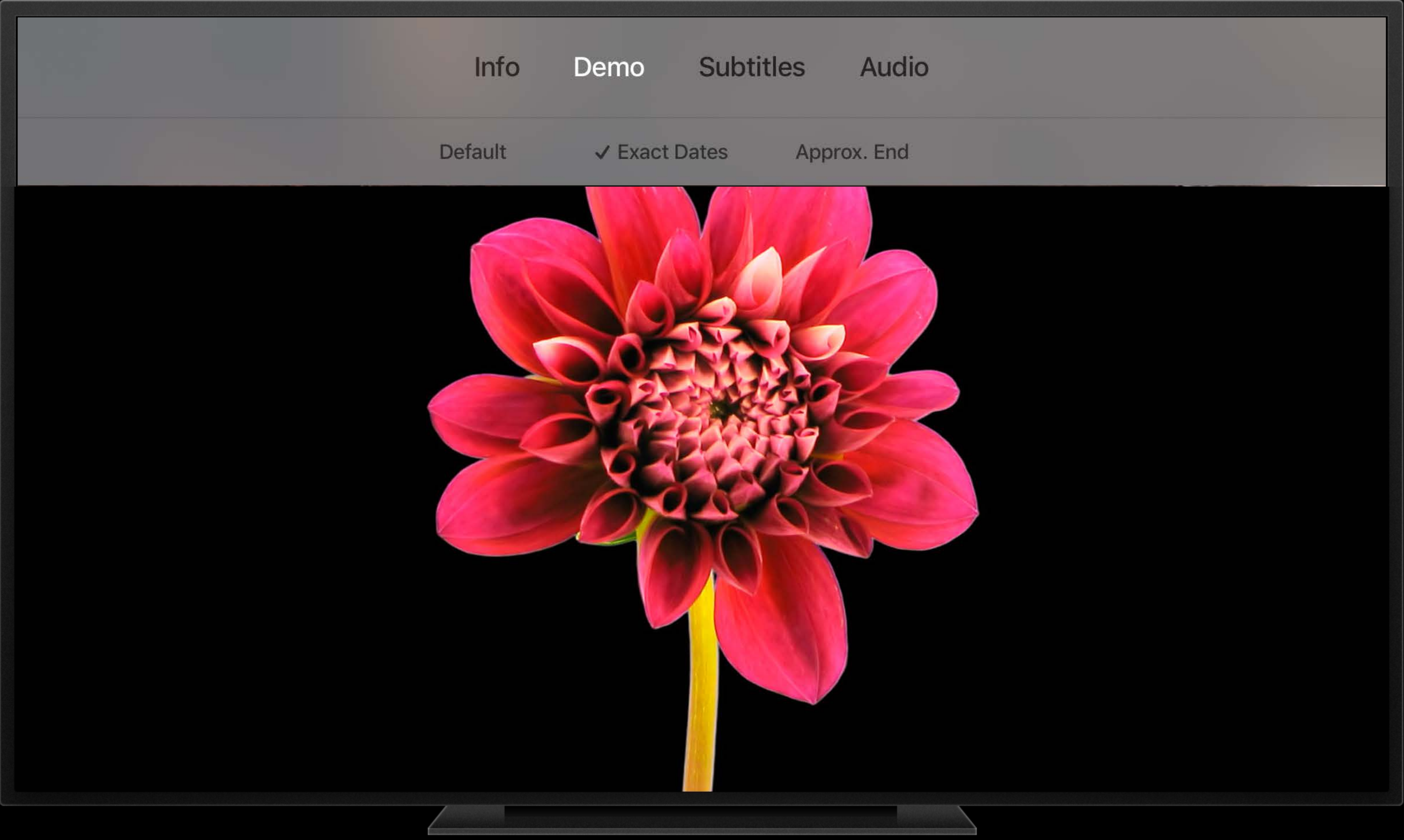


MENU





NEW



# Custom Info View

NEW

Create your own UIViewController

# Custom Info View



NEW

Create your own UIViewController

Client determines the contents

# Custom Info View



NEW

Create your own UIViewController

Client determines the contents

Set the customInfoViewController property

# Custom Info View



NEW

Create your own UIViewController

Client determines the contents

Set the customInfoViewController property

Keep your style consistent with the other info views

# Custom Info View

Guidelines



NEW

# Custom Info View

Guidelines



Use standard font styles

# Custom Info View

## Guidelines



Use standard font styles

Keep the view background transparent



# Custom Info View

## Guidelines



NEW

Use standard font styles

Keep the view background transparent

Indicate desired height via auto-layout or preferredContentSize

# Custom Info View

## Guidelines



NEW

Use standard font styles

Keep the view background transparent

Indicate desired height via auto-layout or preferredContentSize

No taller than half the screen

# Custom Info View

## Guidelines



NEW

Use standard font styles

Keep the view background transparent

Indicate desired height via auto-layout or preferredContentSize

No taller than half the screen

Avoid text fields and buttons

# Custom Info View

## Guidelines



NEW

Use standard font styles

Keep the view background transparent

Indicate desired height via auto-layout or preferredContentSize

No taller than half the screen

Avoid text fields and buttons

Use collection and table views

# New Live Stream Metadata



NEW

## New metadata identifiers

- Exact end date
- Approximate end date
- Start date for live streams

# New Live Stream Metadata



NEW

## New metadata identifiers

- Exact end date
- Approximate end date
- Start date for live streams

Transport bar timescale will reflect these dates

# New Live Stream Metadata



NEW

## New metadata identifiers

- Exact end date
- Approximate end date
- Start date for live streams

Transport bar timescale will reflect these dates

Exact values will be displayed; approximate values will not

# New Live Stream Metadata



NEW

## New metadata identifiers

- Exact end date
- Approximate end date
- Start date for live streams

Transport bar timescale will reflect these dates

Exact values will be displayed; approximate values will not

Orients the user



# Example



Live stream **without** specified start and end dates



# Example



Live stream **without** specified start and end dates



Live stream **with** exact start and exact end dates



# Example



Live stream **without** specified start and end dates



Live stream **with** exact start and exact end dates



Live stream with exact start and **approximate** end date



```
// Live stream metadata example

let oneHour = 3600.0
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
let exactStartDateItem = newMetadataItem(identifier:
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)

let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,
value: exactEndDate as NSDate)
    // ...or use AVKitMetadataIdentifierApproximateEndDate

playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

```
// Live stream metadata example
```

```
let oneHour = 3600.0
```

```
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
```

```
let exactStartDateItem = newMetadataItem(identifier:  
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)
```

```
let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
```

```
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,  
value: exactEndDate as NSDate)
```

```
// ...or use AVKitMetadataIdentifierApproximateEndDate
```

```
playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

```
// Live stream metadata example

let oneHour = 3600.0
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
let exactStartDateItem = newMetadataItem(identifier:
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)

let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,
value: exactEndDate as NSDate)
    // ...or use AVKitMetadataIdentifierApproximateEndDate

playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

```
// Live stream metadata example
```

```
let oneHour = 3600.0
```

```
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
```

```
let exactStartDateItem = newMetadataItem(identifier:  
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)
```

```
let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
```

```
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,  
value: exactEndDate as NSDate)
```

```
// ...or use AVKitMetadataIdentifierApproximateEndDate
```

```
playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

```
// Live stream metadata example

let oneHour = 3600.0
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
let exactStartDateItem = newMetadataItem(identifier:
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)

let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,
value: exactEndDate as NSDate)
    // ...or use AVKitMetadataIdentifierApproximateEndDate

playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```



```
// Live stream metadata example
```

```
let oneHour = 3600.0
```

```
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
```

```
let exactStartDateItem = newMetadataItem(identifier:  
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)
```

```
let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
```

```
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,  
value: exactEndDate as NSDate)
```

```
// ...or use AVKitMetadataIdentifierApproximateEndDate
```

```
playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

```
// Live stream metadata example

let oneHour = 3600.0
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
let exactStartDateItem = newMetadataItem(identifier:
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)

let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,
value: exactEndDate as NSDate)
    // ...or use AVKitMetadataIdentifierApproximateEndDate

playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

```
// Live stream metadata example
```

```
let oneHour = 3600.0
```

```
let exactStartDate = Date(timeIntervalSinceNow: -oneHour)
```

```
let exactStartDateItem = newMetadataItem(identifier:  
AVKitMetadataIdentifierExactStartDate, value: exactStartDate as NSDate)
```

```
let exactEndDate = exactStartDate.addingTimeInterval(3.0*oneHour)
```

```
let exactEndDateItem = newMetadataItem(identifier: AVKitMetadataIdentifierExactEndDate,  
value: exactEndDate as NSDate)
```

```
// ...or use AVKitMetadataIdentifierApproximateEndDate
```

```
playerItem.externalMetadata.append(contentsOf: [exactStartDateItem, exactEndDateItem])
```

***Demo***

# AVKit Summary

Overlay layout guide

Transport bar animation coordinator

Better control of dismissal

Custom info views

New live stream metadata

# AVKit Summary

Overlay layout guide

Transport bar animation coordinator

Better control of dismissal

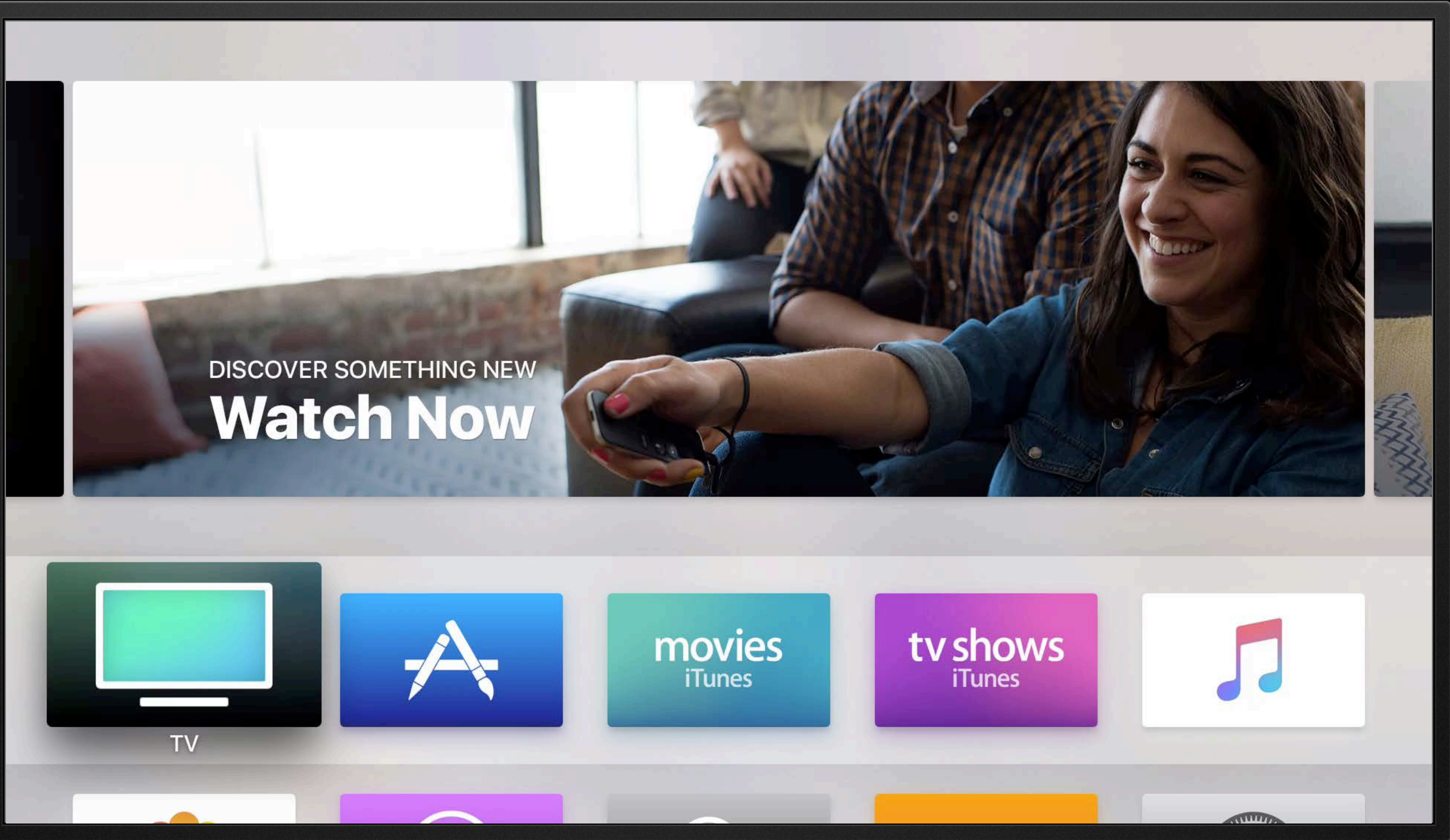
Custom info views

New live stream metadata

# **View Controller Enhancements**

# Modal Presentations



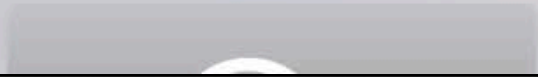


DISCOVER SOMETHING NEW

**Watch Now**



TV



# `.blurOverFullScreen`



NEW

New `UIModalPresentationStyle`

# `.blurOverFullScreen`

NEW

New `UIModalPresentationStyle`

Progressively blurs the presenting view controller

# .blurOverFullScreen



NEW

New `UIModalPresentationStyle`

Progressively blurs the presenting view controller

Uses the same animations and timings as `UIAlertController`

```
viewController.modalPresentationStyle = .blurOverFullScreen  
self.present(viewController, animated: true, completion: nil)
```

```
viewController.modalPresentationStyle = .blurOverFullScreen
self.present(viewController, animated: true, completion: nil)
```

UISplitViewController

## View Controllers

**UIAlertController**  
AlertsViewController

UICollectionViewController  
CollectionViewController

UIPageViewController  
PageViewController

AVPlayerViewController  
VideoPlayerViewController

Show Single Option Alert

Show Multiple Option Alert

Show Destructive Option Alert



## View Controllers

Show Single Option Alert

Show Multiple Option Alert

Show Destructive Option Alert

**UIAlertController**  
AlertsViewController

UICollectionViewController  
CollectionViewController

UIPageViewController  
PageViewController

AVPlayerViewController  
VideoPlayerViewController

NEW

```
public enum UISplitViewControllerPrimaryEdge : Int {  
    case leading  
    case trailing  
}
```



NEW

```
public enum UISplitViewControllerPrimaryEdge : Int {  
    case leading  
    case trailing  
}  
  
open var primaryEdge: UISplitViewControllerPrimaryEdge
```

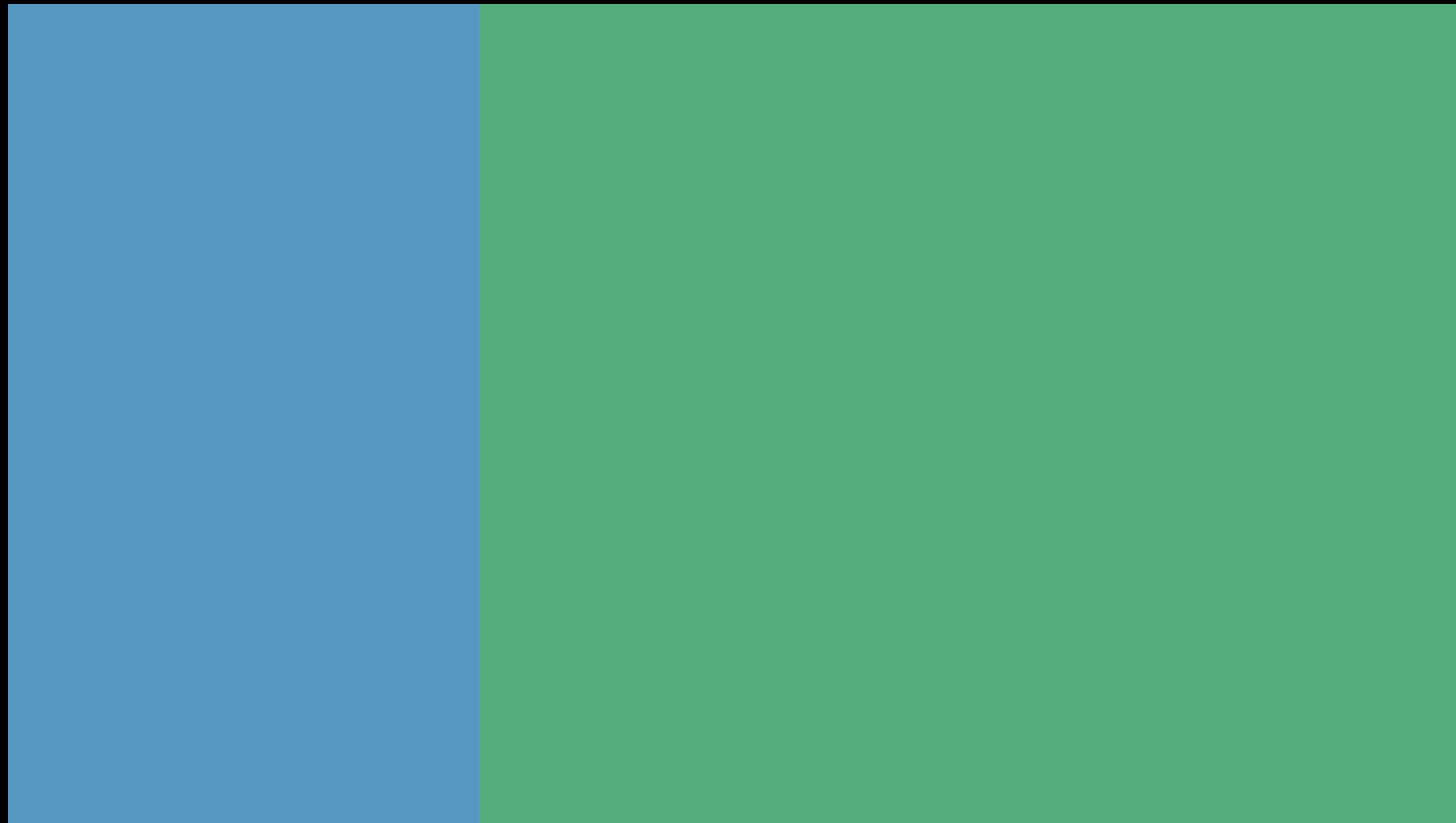


NEW

.leading

.trailing

LTR



RTL



.leading

.trailing

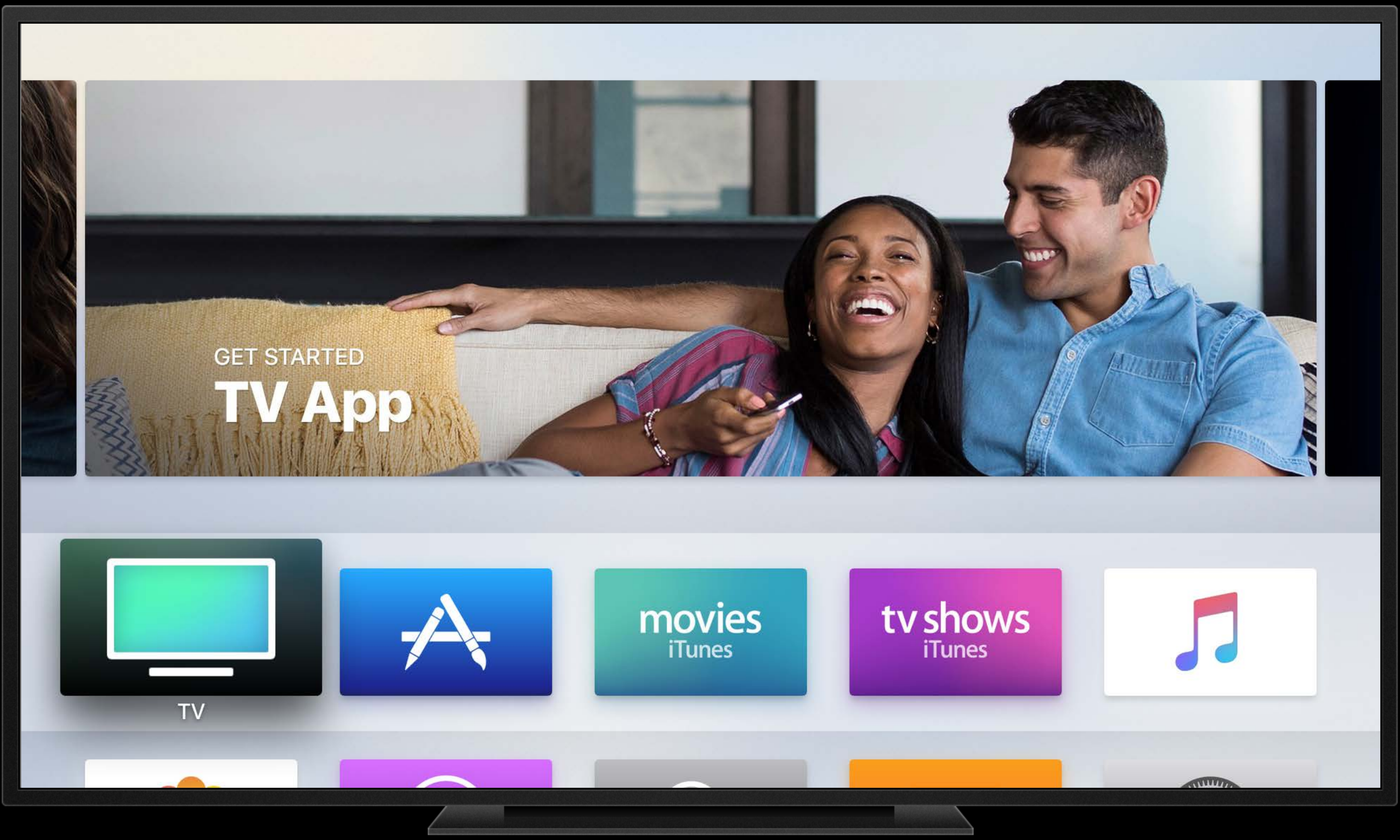
LTR



RTL



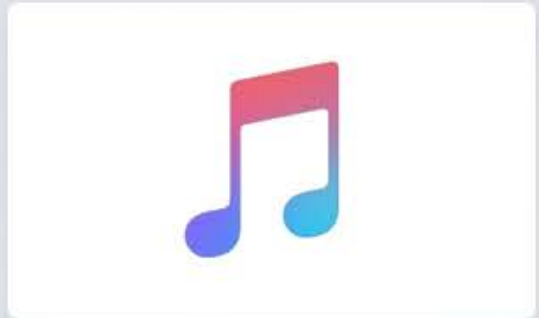
Light/Dark Appearance



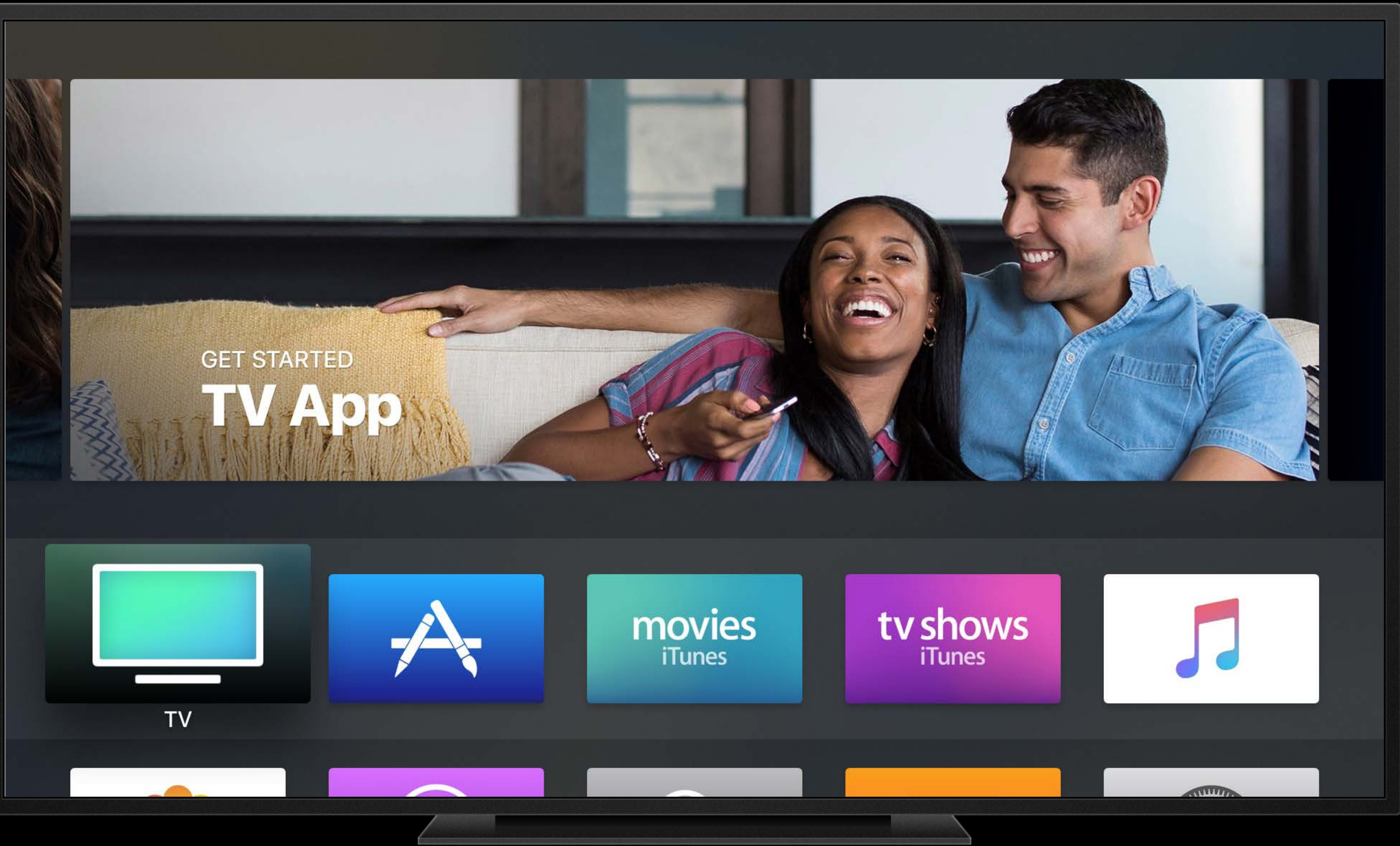
GET STARTED  
**TV App**



TV







GET STARTED  
**TV App**



TV



## Appearance



Light

Dark

Automatic ✓

Select Light or Dark to change the appearance of backgrounds and menus. Select Automatic to change the appearance to Dark at night, and Light during the day.

## Appearance



Light

Dark

Automatic ✓

Select Light or Dark to change the appearance of backgrounds and menus. Select Automatic to change the appearance to Dark at night, and Light during the day.

## Appearance



Light

Dark

Automatic ✓

Select Light or Dark to change the appearance of backgrounds and menus. Select Automatic to change the appearance to Dark at night, and Light during the day.

**Top Shelf**

Weather

San Jose



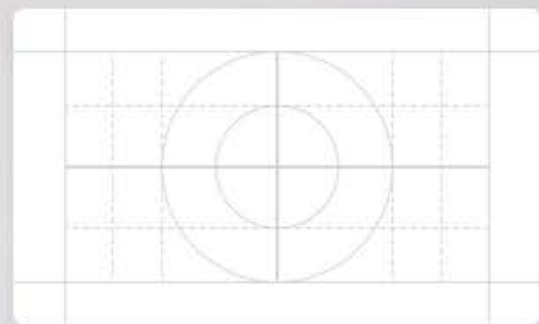
San Francisco



Santa Cruz



Sa



Weather

San Jose



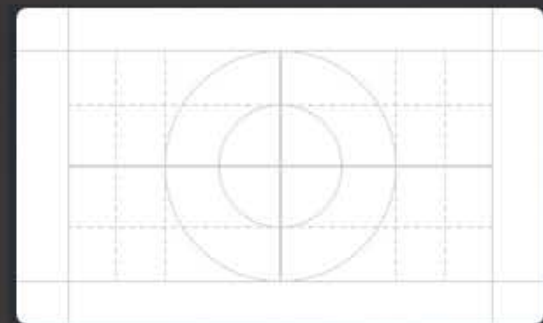
San Francisco



Santa Cruz



Sa



```
let contentIdentifier = TVContentIdentifier(identifier: key, container: nil)
```

```
let contentItem = TVContentItem(contentIdentifier: contentIdentifier!)
```

```
contentItem?.imageURL = imageURL
```



```
let contentIdentifier = TVContentIdentifier(identifier: key, container: nil)
```

```
let contentItem = TVContentItem(contentIdentifier: contentIdentifier!)
```

```
contentItem?.setImageURL(lightImageURL, forTraits: .userInterfaceStyleLight)
```

```
contentItem?.setImageURL(darkImageURL, forTraits: .userInterfaceStyleDark)
```

Adapting For Your Content

# HIDDEN FIGURES

GENRE 93% 10+ 2 hr 7 min 2016 **PG** **HD** **CC** **SDH** **AD**  
Drama

DIRECTOR  
Theodore Melfi

STARRING  
Taraji P. Henson  
Octavia Spencer  
Janelle Monáe  
Kevin Costner  
Kirsten Dunst

The incredible untold true story of Katherine Johnson (Taraji P. Henson), Dorothy Vaughan (Octavia Spencer) & Mary Jackson (Janelle Monáe)—brilliant African-American women working at NASA, who served as the brains behind one of the greatest operations in history: the launch of astronau... [MORE](#)



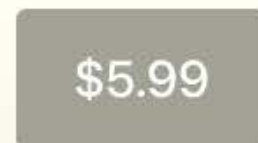
Preview



\$9.99

Buy

With iTunes Extras



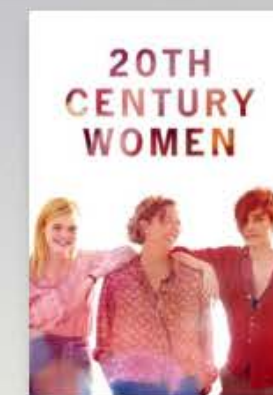
\$5.99

Rent



Wish List

## More in Drama



MARVEL STUDIOS  
**GUARDIANS OF THE GALAXY**  
VOL. 2

GENRE  
Action & Adventure

82% 12+ PG-13 HD

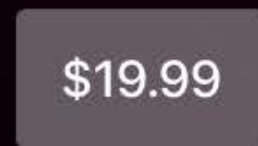
DIRECTOR  
James Gunn

Set to the sonic backdrop of Awesome Mixtape #2, Marvel Studios' Guardians of the Galaxy Vol. 2 continues the team's adventures as they traverse the outer reaches of the cosmos. The Guardians must fight to keep their newfound family together as they unravel the mystery of Peter Quill's t... MORE

STARRING  
Chris Pratt  
Zoe Saldana  
Dave Bautista  
Vin Diesel  
Bradley Cooper



Preview

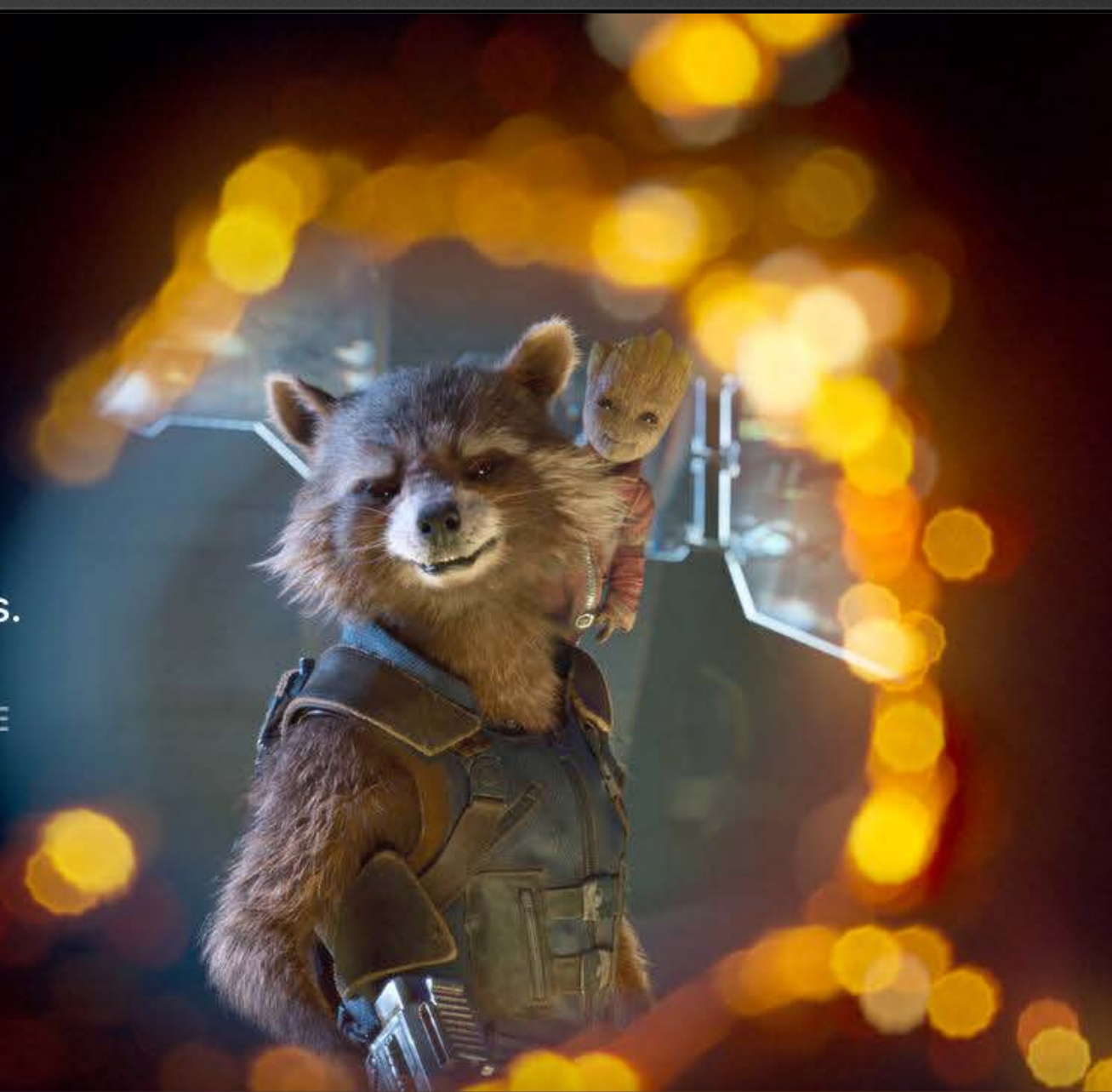


\$19.99

Pre-Order



Wish List



More in Action & Adventure



# preferredUserInterfaceStyle



NEW

Used to specify the `UIUserInterfaceStyle` for a view controller

# preferredUserInterfaceStyle

NEW

Used to specify the `UIUserInterfaceStyle` for a view controller

Allows you to customize the appearance to match your content

# preferredUserInterfaceStyle



NEW

Used to specify the `UIUserInterfaceStyle` for a view controller

Allows you to customize the appearance to match your content

Will automatically adjust the wallpaper to match the style

Light Dark

UIUserInterfaceStyleLight



Light Dark

UIUserInterfaceStyleLight

```
// UIViewController.h
```

```
open var preferredUserInterfaceStyle: UIUserInterfaceStyle { get }
```

```
open var childViewControllerForUserInterfaceStyle: UIViewController? { get }
```

```
open func setNeedsUserInterfaceAppearanceUpdate()
```

```
// UIViewController.h
```

```
open var preferredUserInterfaceStyle: UIUserInterfaceStyle { get }
```

```
open var childViewControllerForUserInterfaceStyle: UIViewController? { get }
```

```
open func setNeedsUserInterfaceAppearanceUpdate()
```

```
// UIViewController.h
```

```
open var preferredUserInterfaceStyle: UIUserInterfaceStyle { get }
```

```
open var childViewControllerForUserInterfaceStyle: UIViewController? { get }
```

```
open func setNeedsUserInterfaceAppearanceUpdate()
```

```
// UIViewController.h
```

```
open var preferredUserInterfaceStyle: UIUserInterfaceStyle { get }
```

```
open var childViewControllerForUserInterfaceStyle: UIViewController? { get }
```

```
open func setNeedsUserInterfaceAppearanceUpdate()
```

# **View Controller Enhancements**

# Summary

On-Demand Resources bundle size has increased from 200MB to 4GB

# Summary

On-Demand Resources bundle size has increased from 200MB to 4GB

tvOS now supports RTL languages



# Summary

On-Demand Resources bundle size has increased from 200MB to 4GB

tvOS now supports RTL languages

New ways to do work in the background

# Summary

On-Demand Resources bundle size has increased from 200MB to 4GB

tvOS now supports RTL languages

New ways to do work in the background

New AVKit enhancements including custom info views

# Summary

On-Demand Resources bundle size has increased from 200MB to 4GB

tvOS now supports RTL languages

New ways to do work in the background

New AVKit enhancements including custom info views

Image overlays on UIImageView

# Summary

On-Demand Resources bundle size has increased from 200MB to 4GB

tvOS now supports RTL languages

New ways to do work in the background

New AVKit enhancements including custom info views

Image overlays on UIImageView

Adapt to your content using preferredUserInterfaceStyle

# More Information

<https://developer.apple.com/wwdc17/209>

# Related Sessions

---

[Deep Linking on tvOS](#)

WWDC 2017 Video

---

[Now Playing and Remote Commands on tvOS](#)

WWDC 2017 Video

---

What's New in Cocoa Touch

Tuesday 10:20AM

---

Advances in TVMLKit

Tuesday 11:20AM

---

Updating Your App for [iOS 11]

Tuesday 4:10PM

---

[Best Practices and What's New in User Notifications](#)

Hall 2

Wednesday 4:10PM

---

[Focus Interaction in tvOS 11](#)

Grand Ballroom A

Thursday 9:00AM

---

# Labs

---

tvOS Lab

Technology Lab I

Tue 12:00PM–1:50PM

---

tvOS Lab

Technology Lab H

Wed 11:00AM–2:00PM

---

tvOS Lab

Technology Lab I

Thur 11:00AM–1:00PM

---

AVKit Lab

Technology Lab G

Tue 1:00PM–4:10PM

---

AVKit Lab

Technology Lab F

Wed 11:00AM–1:00PM

---

