

# What's New with User Notifications

*(or Notifications without being Pushy)*

CocoaConf | Washington DC | September 2016

Kevin Y. Kim | @kykim | [projector.com](http://projector.com)

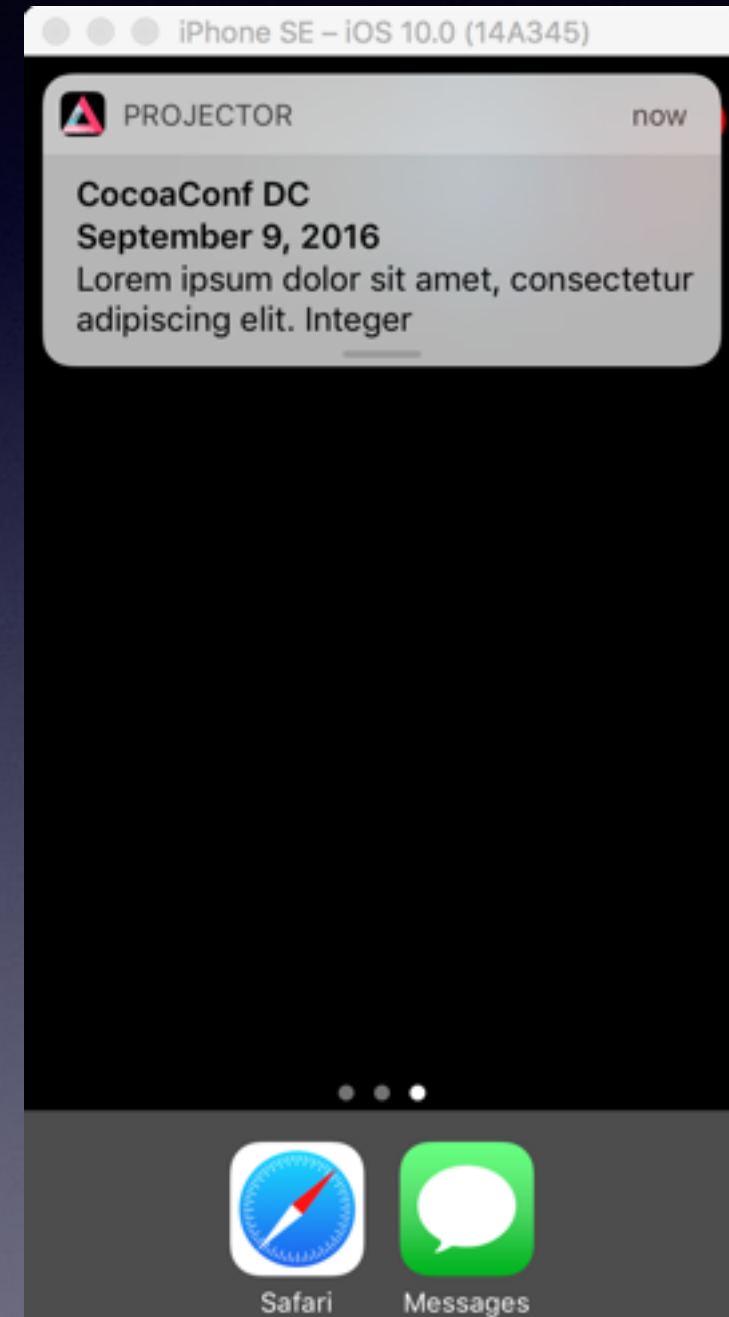
# Agenda

- What are User Notifications?
- Life Before iOS 10
- What's Changed in iOS 10
- Service and Content Extensions

# What are User Notifications?

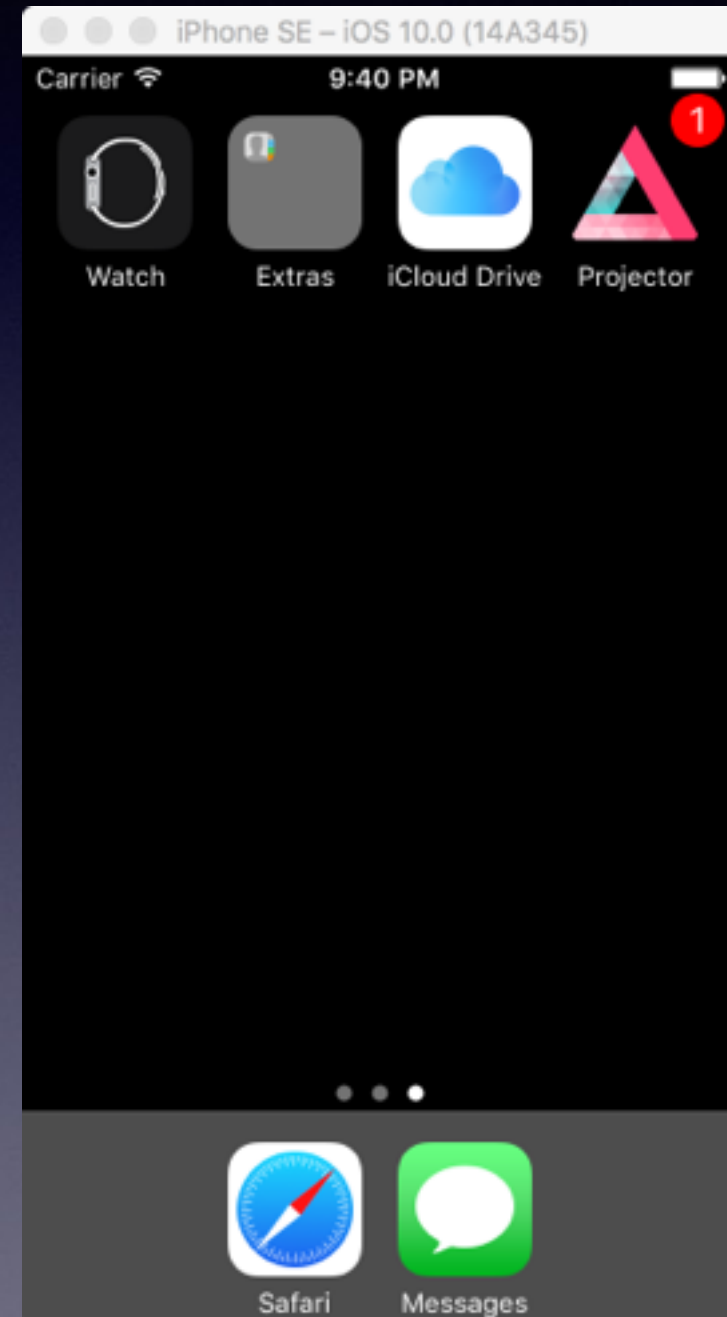
# What are User Notifications?

- Alerts



# What are User Notifications?

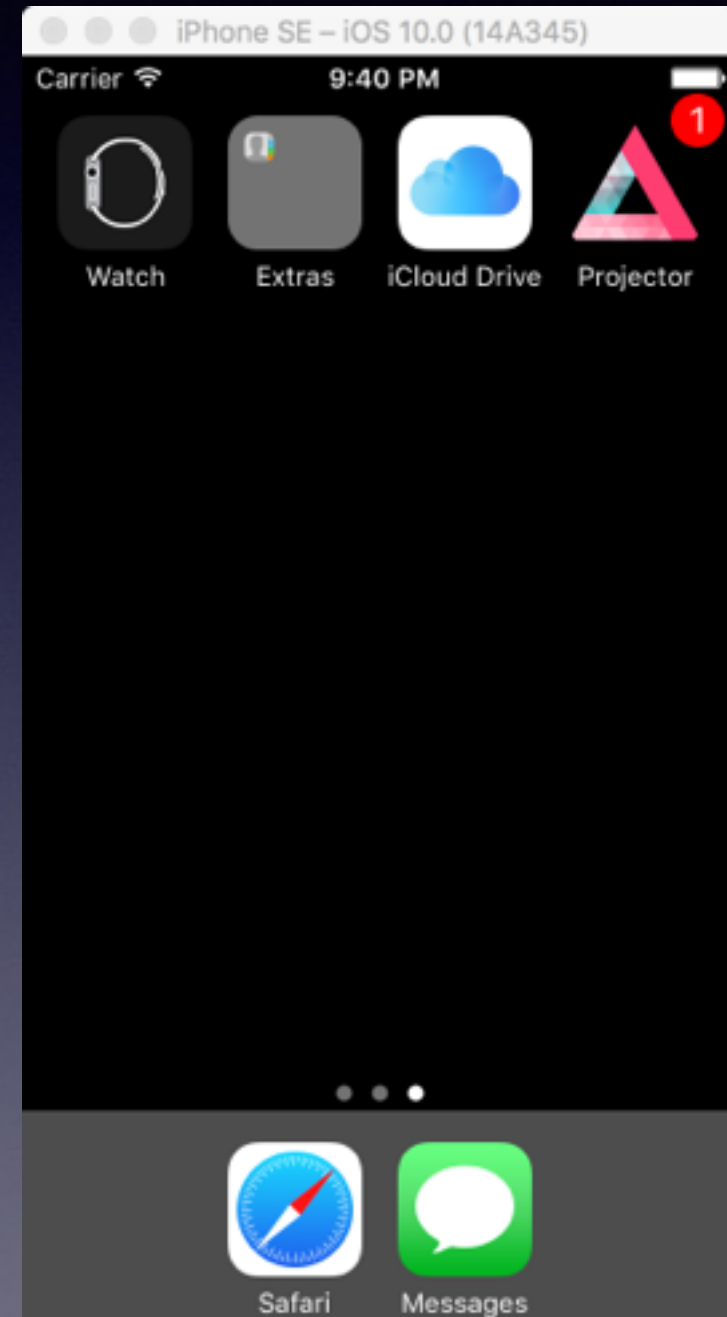
- Alerts
- Badge





# What are User Notifications?

- Alerts
- Badge
- Sound



# Types of Notifications?

# Types of Notifications?

- Local Notifications



# Types of Notifications?

- Local Notifications
- Remote (or Push) Notifications

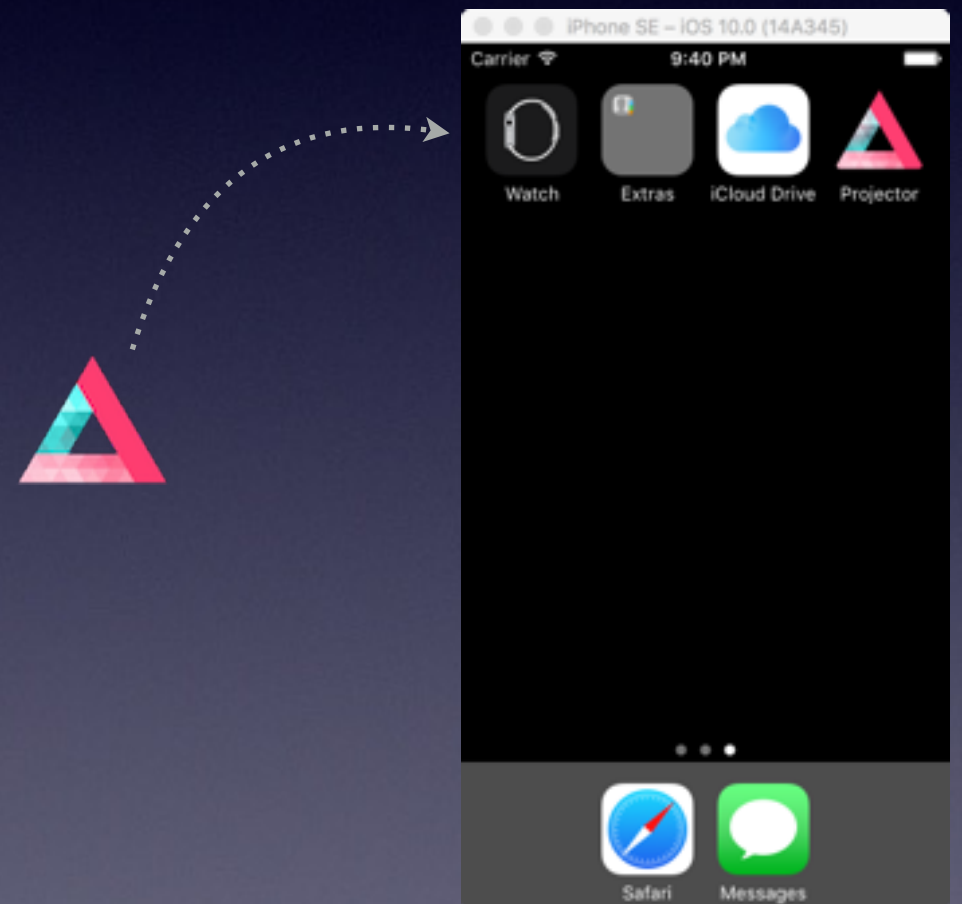
# Types of Notifications?

- Local Notifications
- Remote (or Push) Notifications
- Silent (or Background) Notifications

# Local Notifications

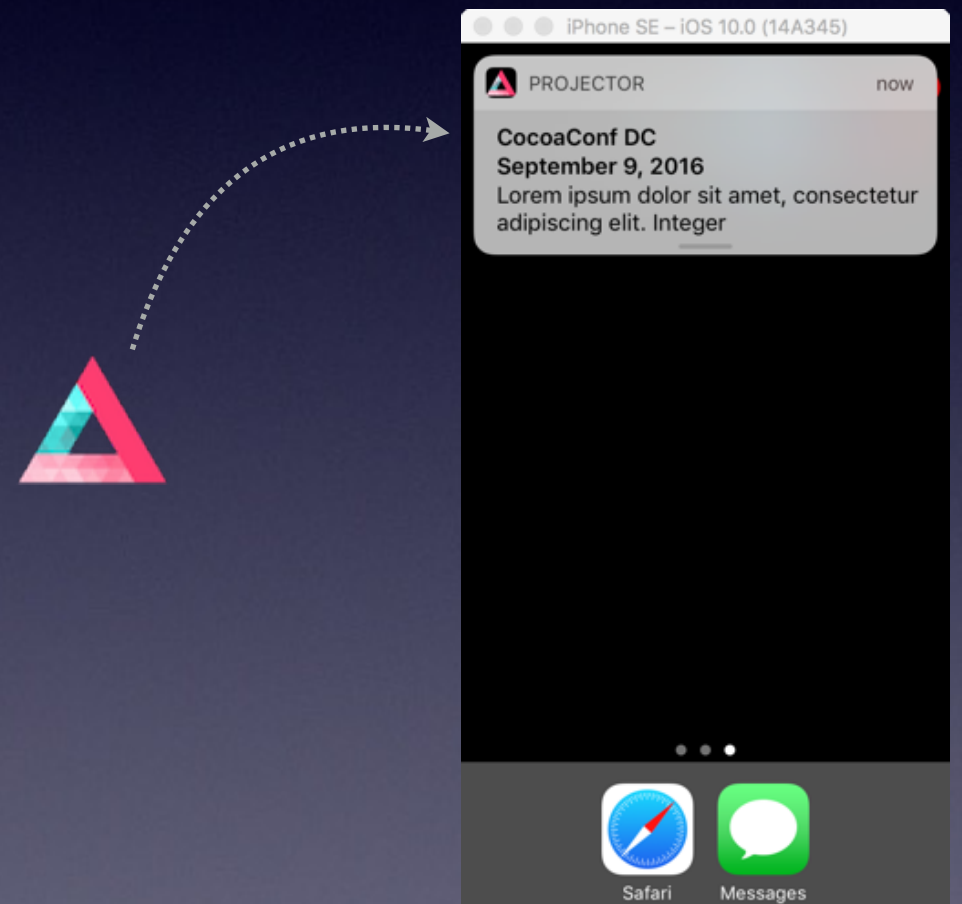
# Local Notifications

- Application creates Notification



# Local Notifications

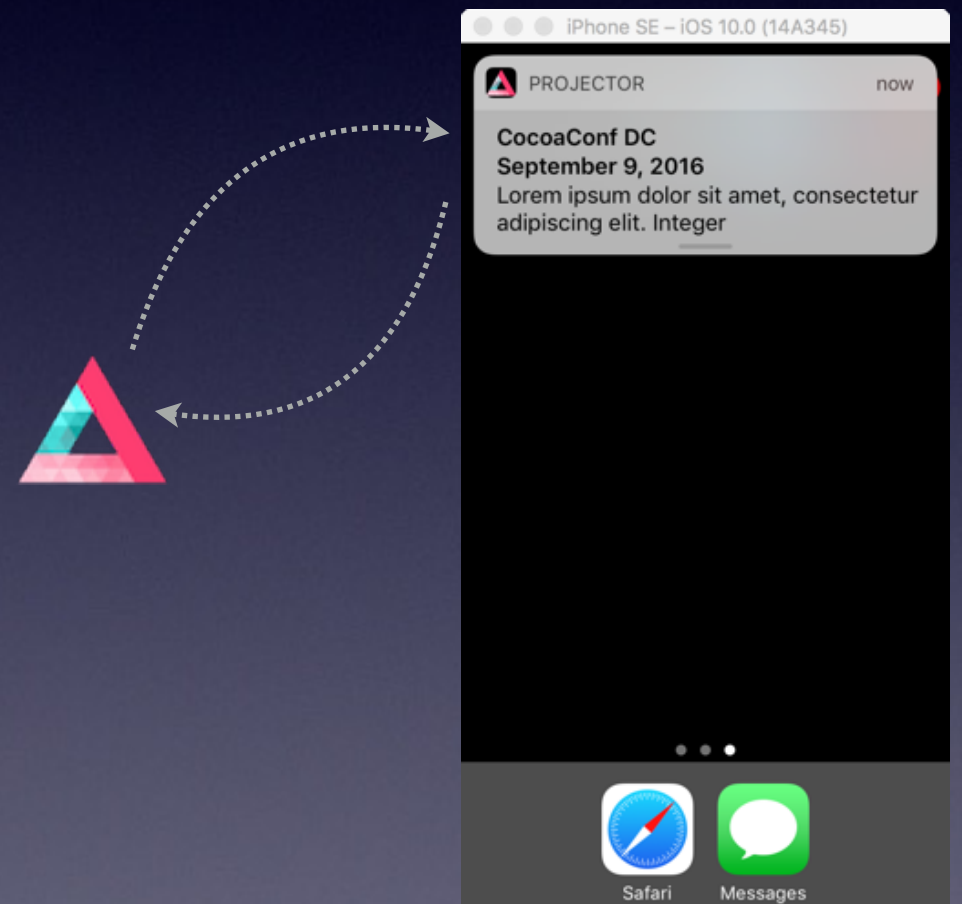
- Application creates Notification
  - Calendar Trigger
  - Location Trigger
  - Time Interval Trigger





# Local Notifications

- Application creates Notification
  - Calendar Trigger
  - Location Trigger
  - Time Interval Trigger
- Application receives User Action

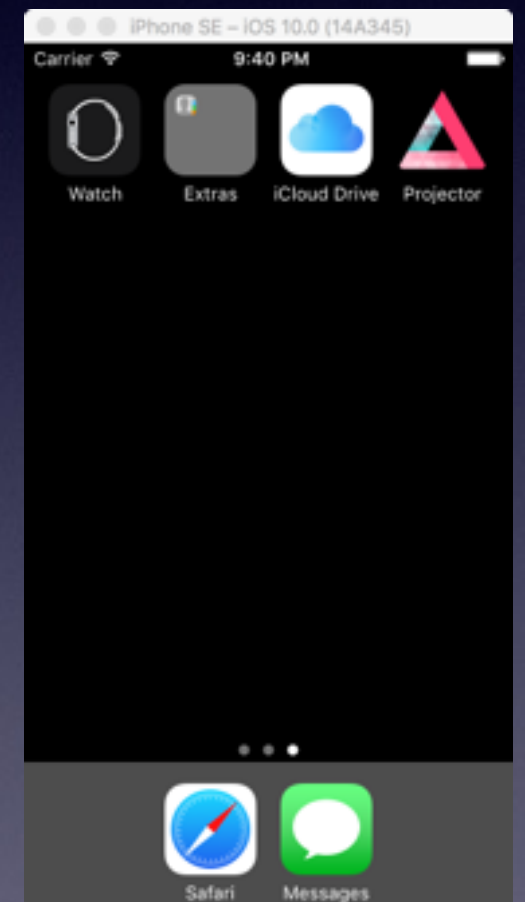


# Remote Notifications

# Remote Notifications

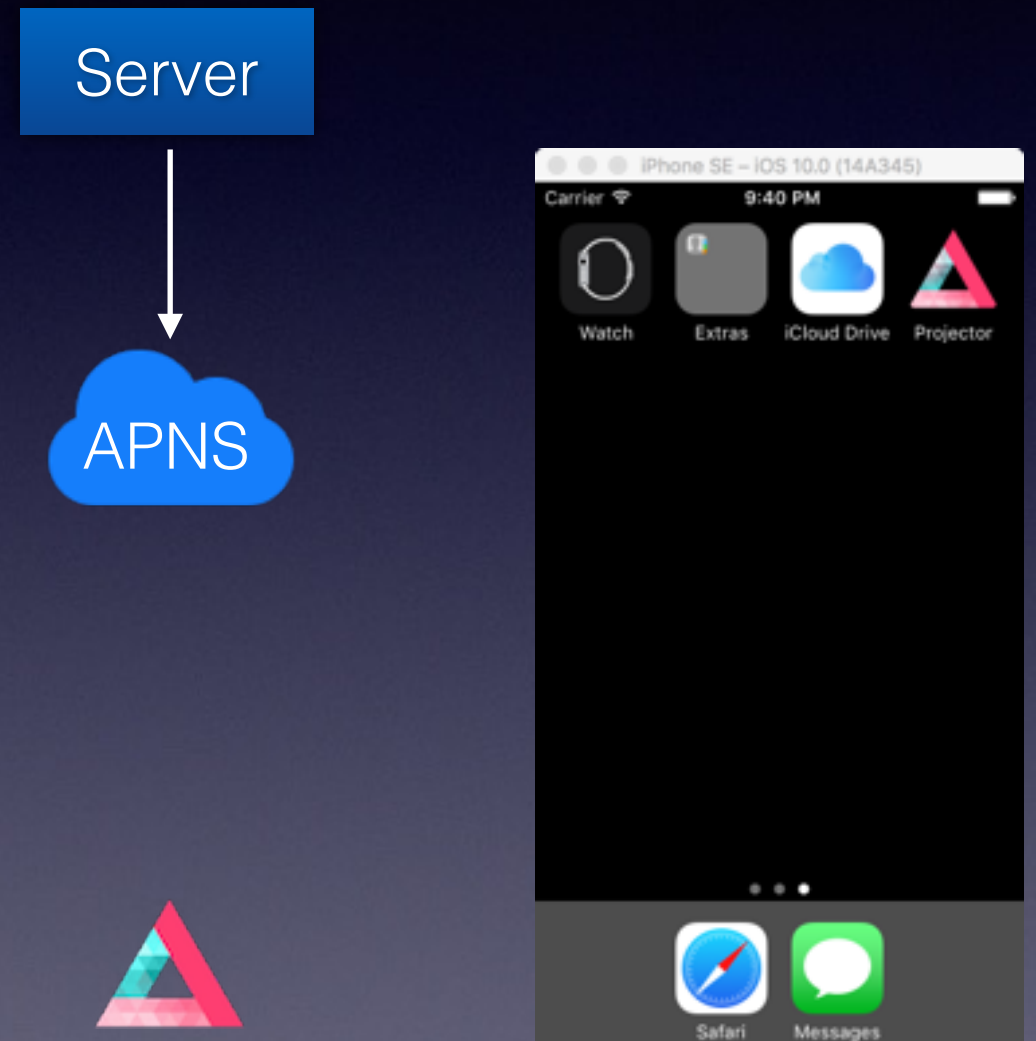
Server

APNS



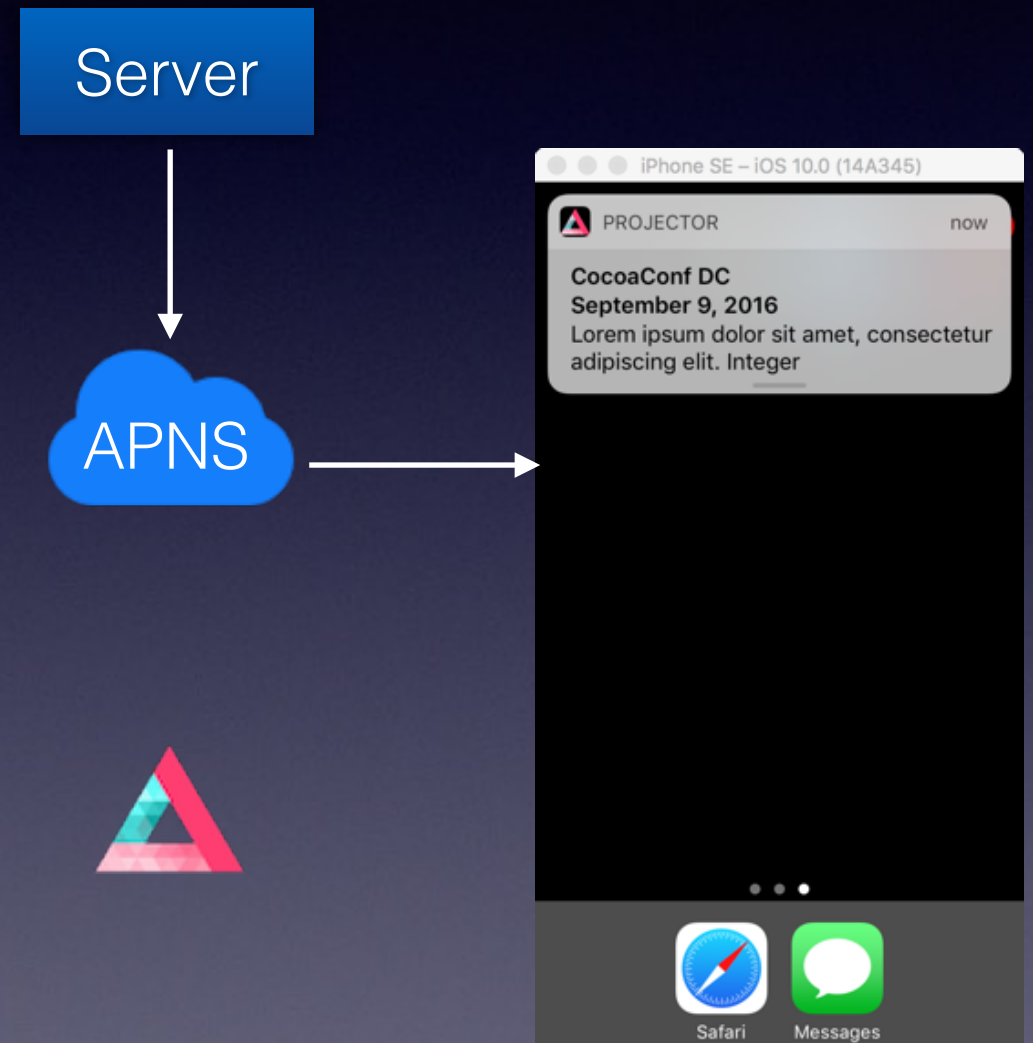
# Remote Notifications

- Server sends payload to APNS



# Remote Notifications

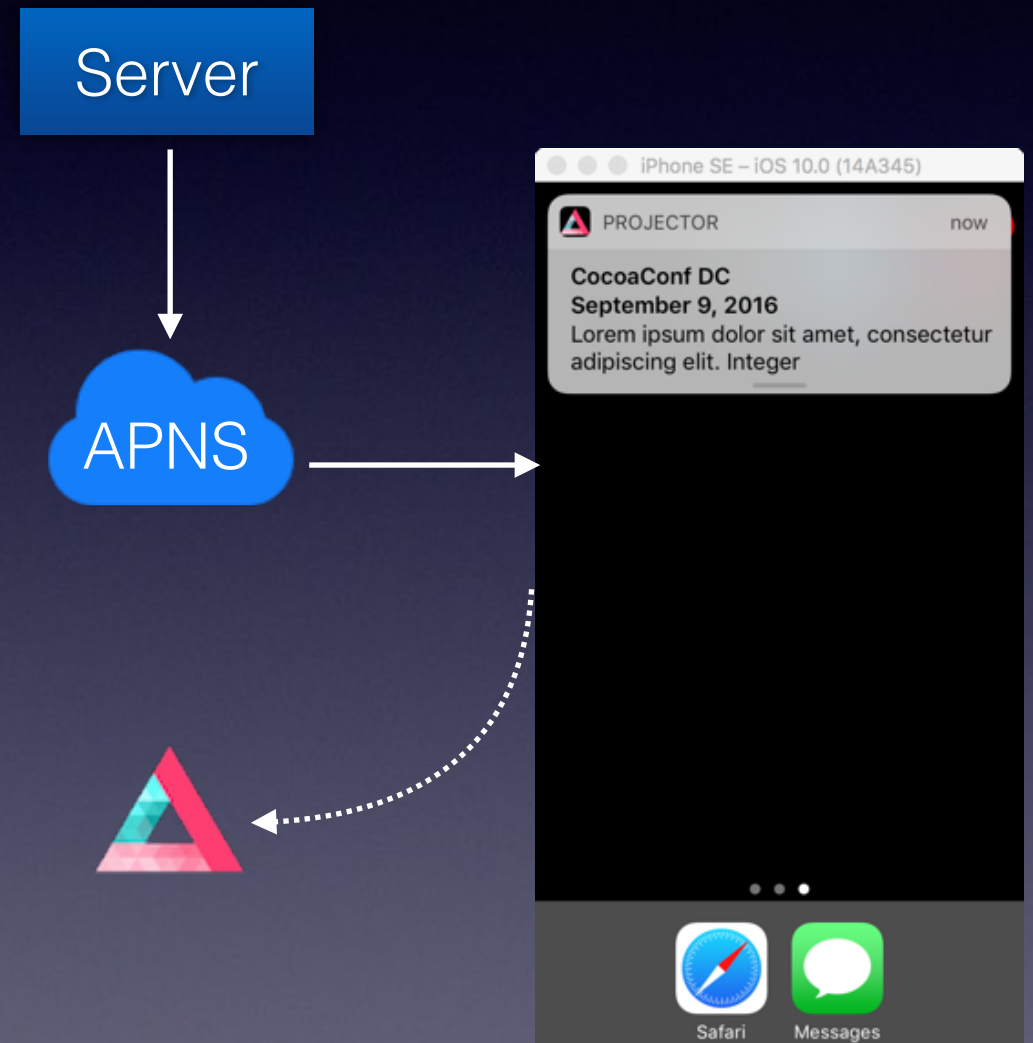
- Server sends payload to APNS
- APNS sends Notification to Device





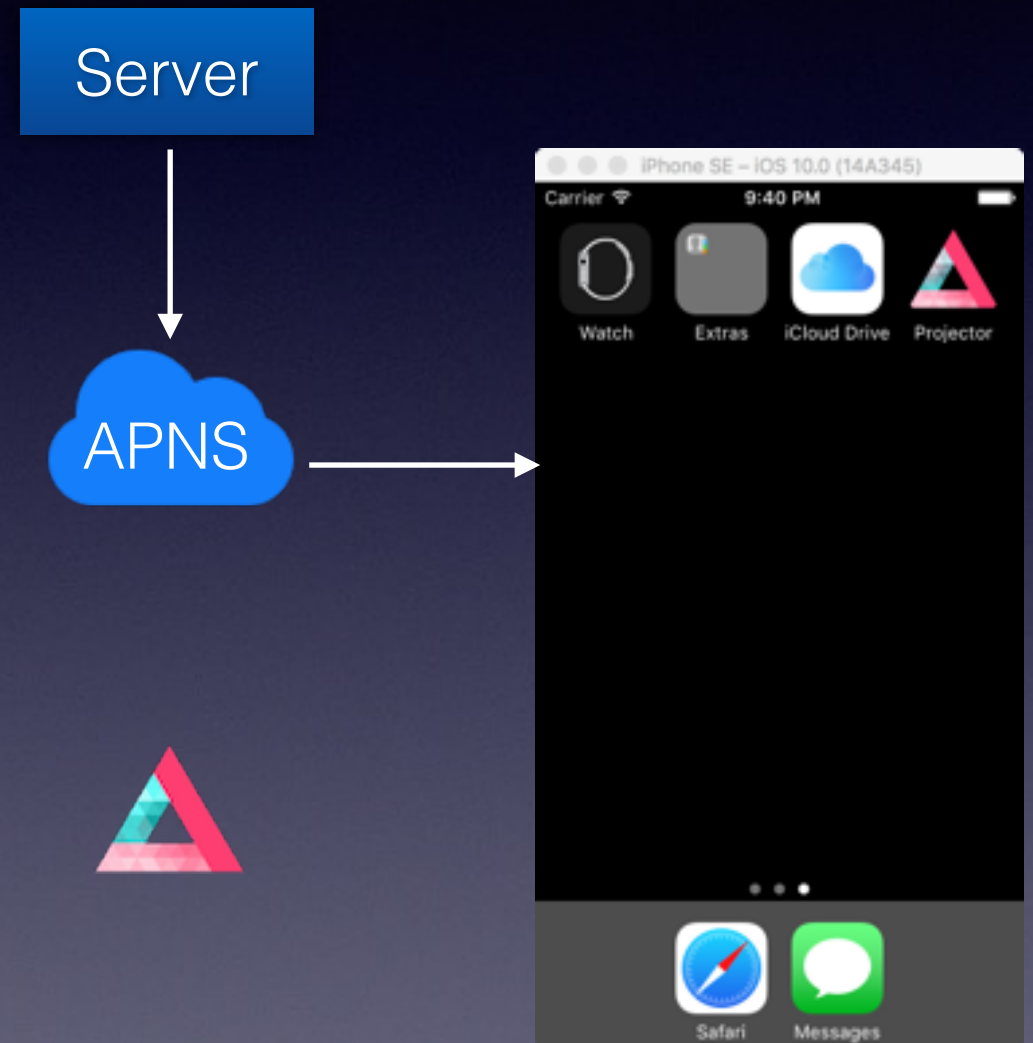
# Remote Notifications

- Server sends payload to APNS
- APNS sends Notification to Device
- Application receives User Action



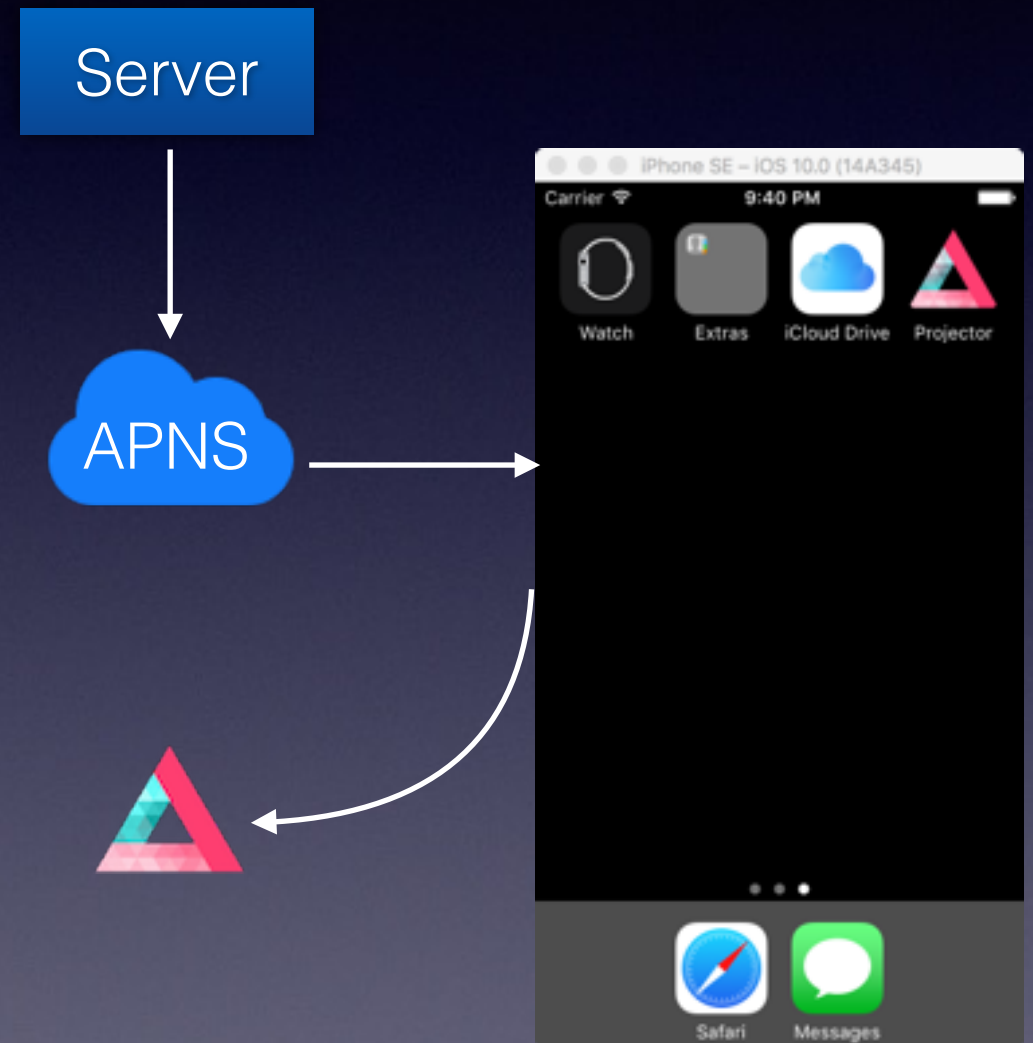
# Silent Notifications

- content-available: 1



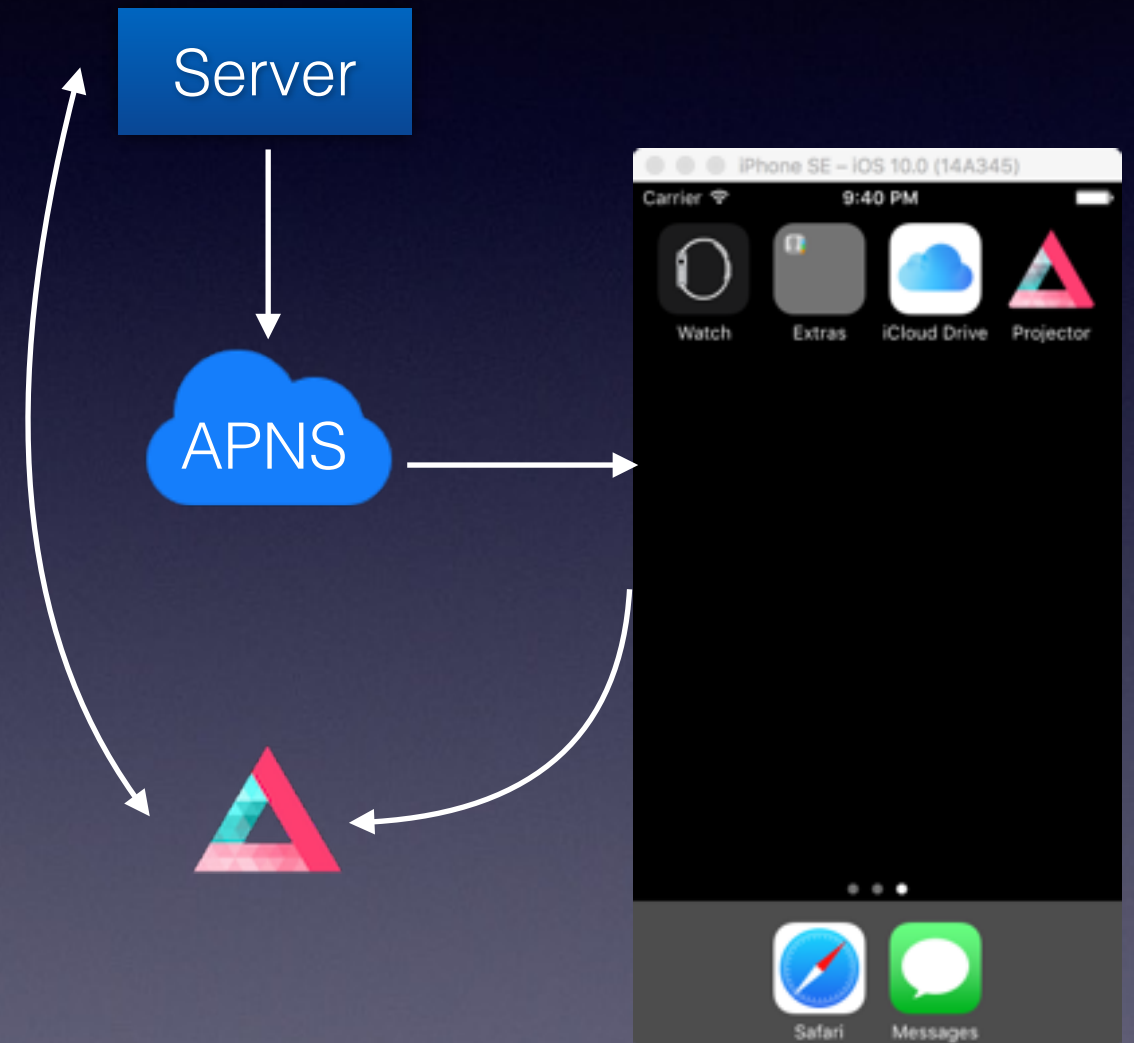
# Silent Notifications

- content-available: 1
- Wakes Application in Background



# Silent Notifications

- content-available: 1
- Wakes Application in Background
- Refresh Data from Server



# Old Way of Notifications



# Old Way of Notifications

*Pre iOS 10*

# Old Way of Notifications

- Register and Schedule via UIApplication

# Old Way of Notifications

- Register and Schedule via UIApplication
- Different Classes for Local & Remote Notifications

# Old Way of Notifications

- Register and Schedule via UIApplication
- Different Classes for Local & Remote Notifications
- Different Callbacks for Local & Remote Actions



# Old Way of Notifications

UIApplication

- .registerUserNotificationSettings(\_:)
- .registerForRemoteNotifications()

UIApplicationDelegate

- .application(\_:didRegisterUserNotificationSettings:)
- .application(\_:didRegisterForRemoteNotificationsWithDeviceToken:)
- .application(\_:didFailToRegisterForRemoteNotificationsWithError:)

UIApplication

- .scheduleLocalNotification(\_:)
- .presentLocalNotificationNow(\_:)
- .cancelLocalNotification(\_:)
- .cancelAllLocalNotifications()
- .scheduledLocalNotifications()



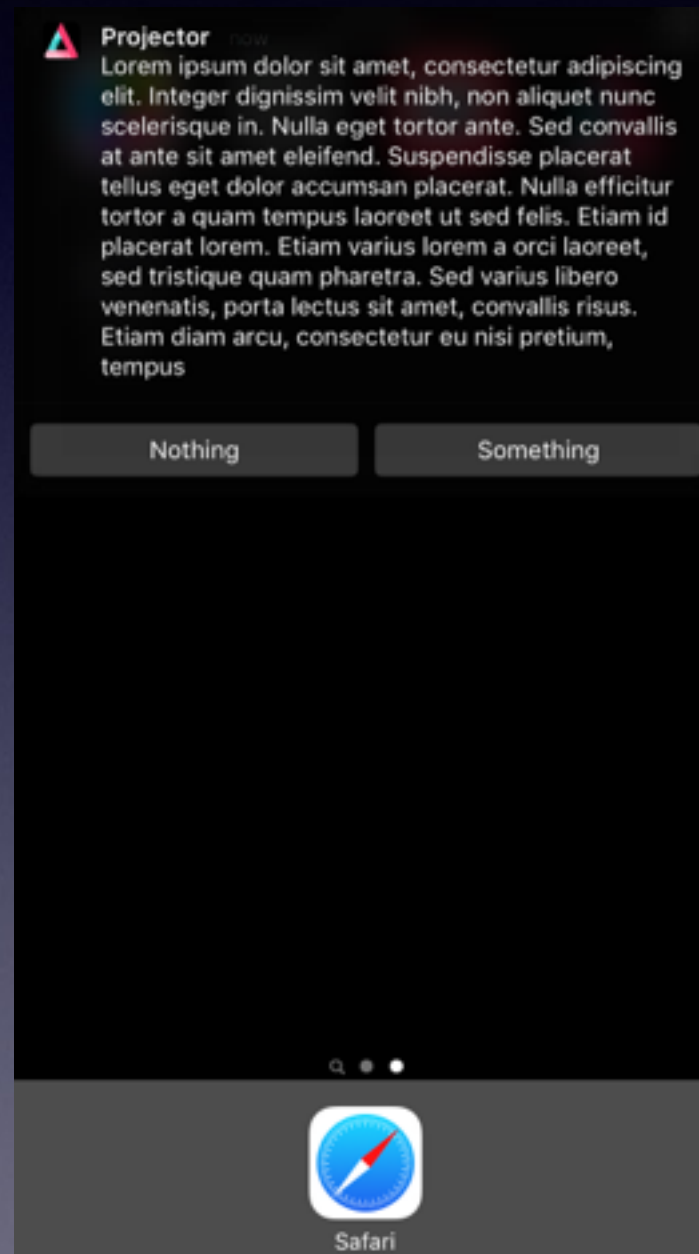
# Old Way of Notifications

UIApplicationDelegate

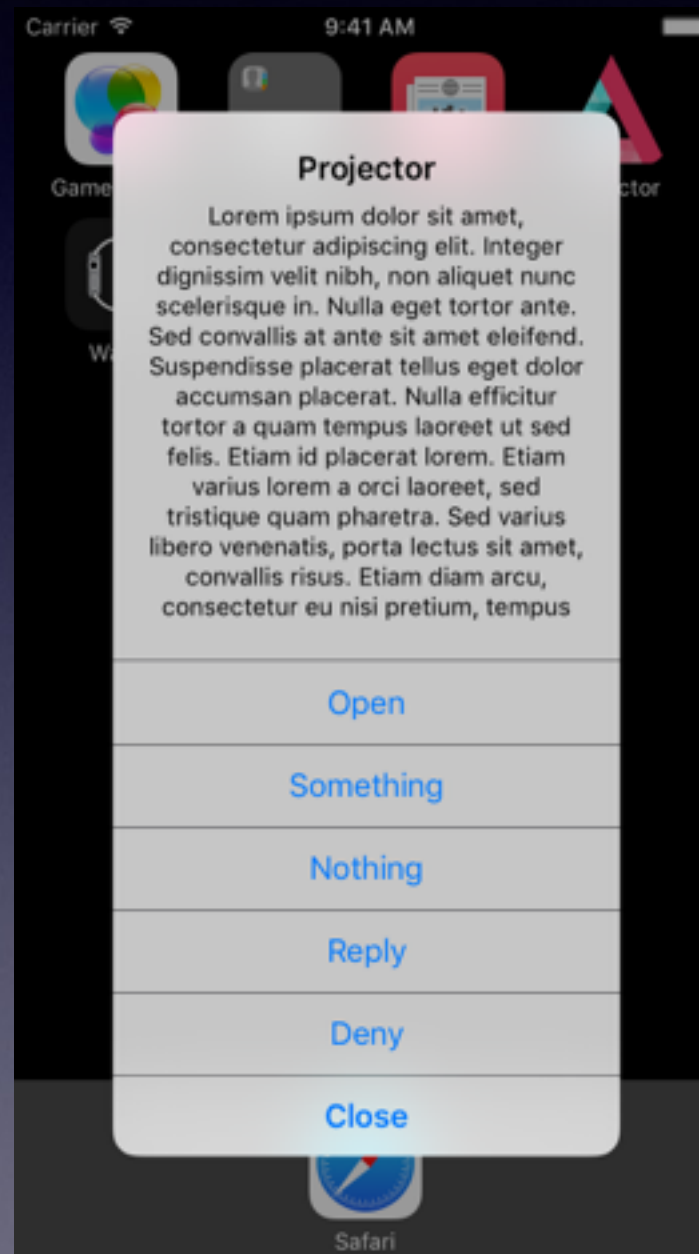
```
.application(_:didReceiveLocalNotification:)  
.application(_:handleActionWithIdentifier:forLocalNotification:  
    completionHandler:)  
.application(_:handleActionWithIdentifier:forLocalNotification:withResponseInfo:  
    completionHandler:)  
  
.application(_:didReceiveRemoteNotification:fetchCompletionHandler:)  
.application(_:handleActionWithIdentifier:forRemoteNotification:  
    completionHandler:)  
.application(_:didReceiveRemoteNotification:)  
.application(_:handleActionWithIdentifier:forRemoteNotification:withResponseInfo:  
    completionHandler:)
```

# Categories & Actions

# Categories & Actions

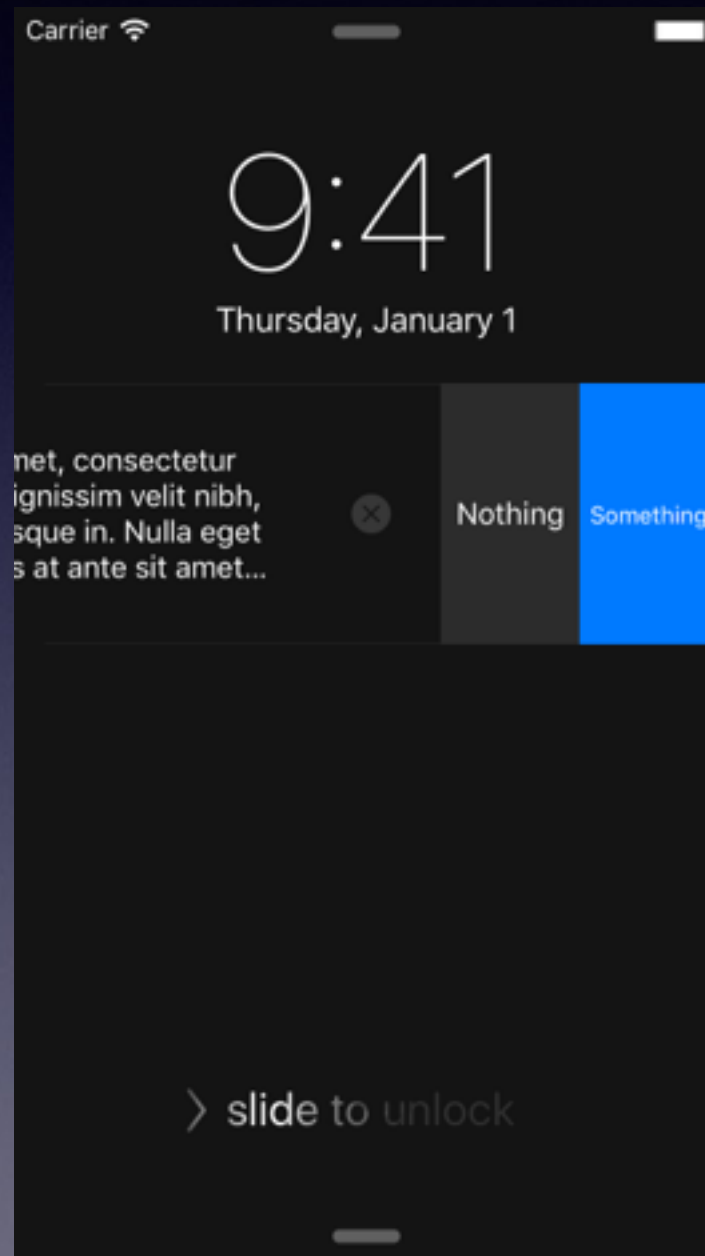


# Categories & Actions



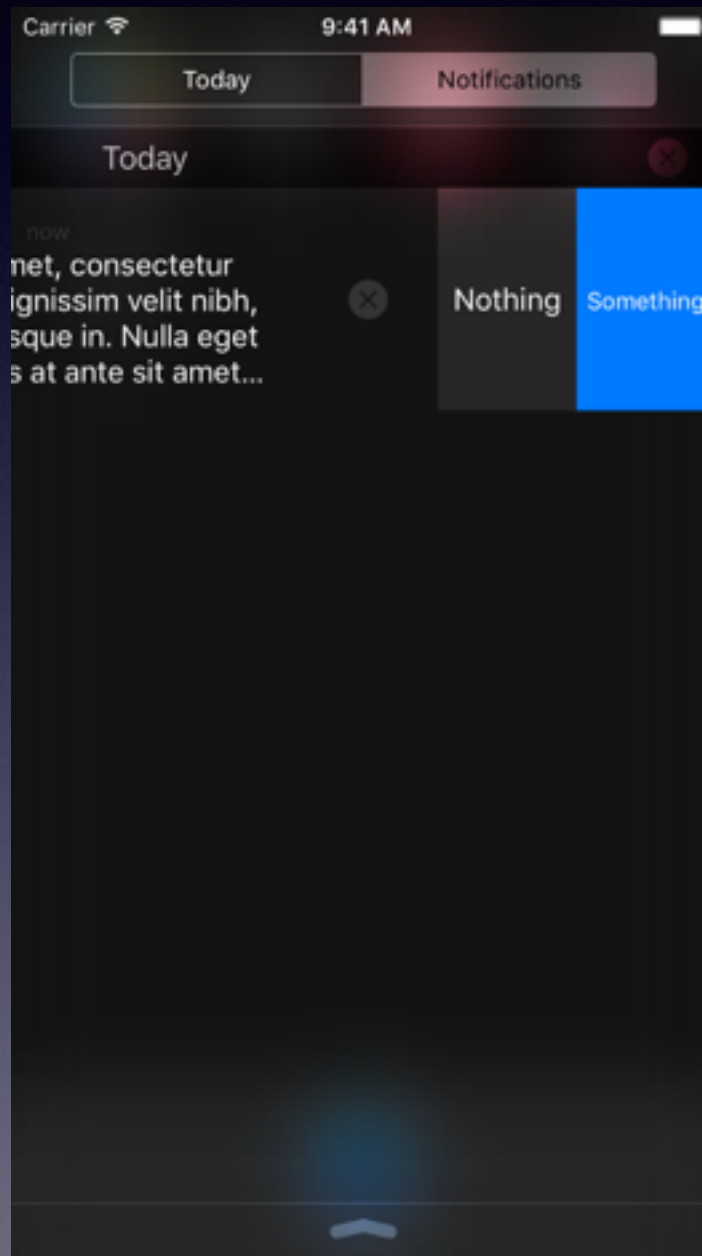


# Categories & Actions





# Categories & Actions



# Categories & Actions

```
let action = UIMutableUserNotificationAction()  
action.identifier = "action-identifier"  
action.title = "Action"  
action.activationMode = UIUserNotificationActivationMode.Background  
action.authenticationRequired = false  
action.destructive = false
```

# Categories & Actions

```
let action = UIMutableUserNotificationAction()  
action.identifier = "action-identifier"  
action.title = "Action"  
action.activationMode = UIUserNotificationActivationMode.Foreground  
action.authenticationRequired = false  
action.destructive = false
```

# Categories & Actions

```
let action = UIMutableUserNotificationAction()  
action.identifier = "action-identifier"  
action.title = "Action"  
action.activationMode = UIUserNotificationActivationMode.Foreground  
action.authenticationRequired = true  
action.destructive = false
```

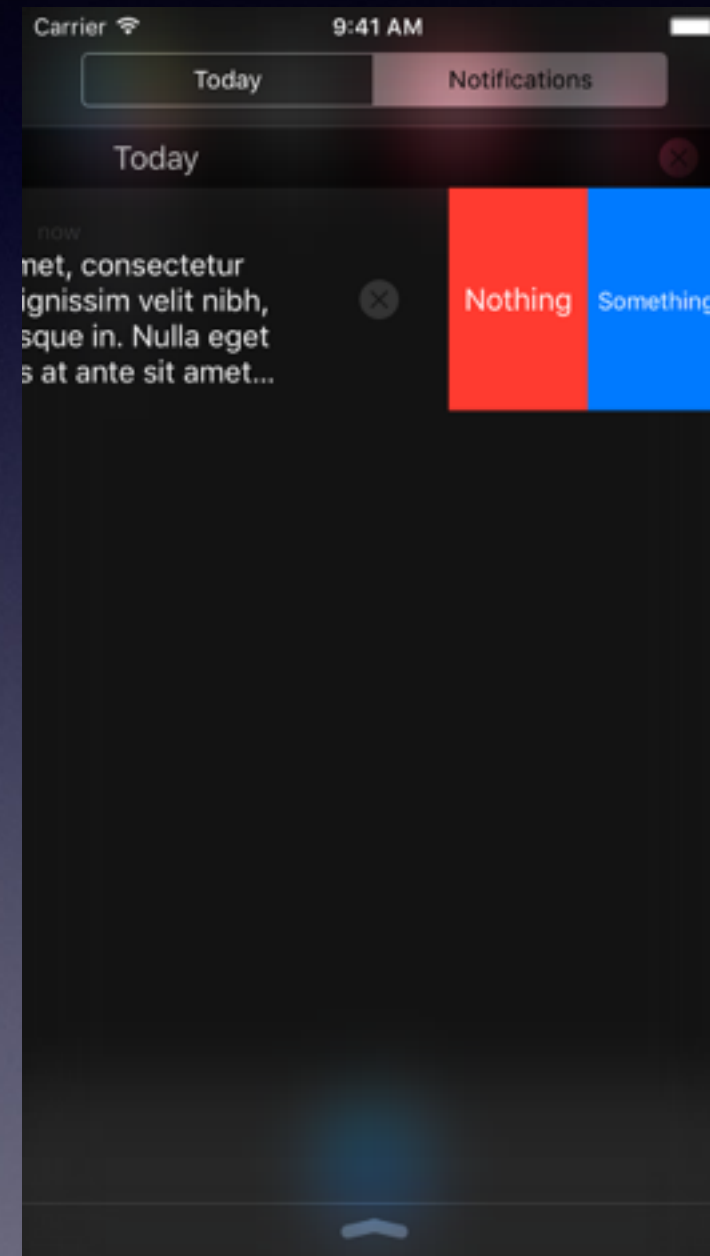
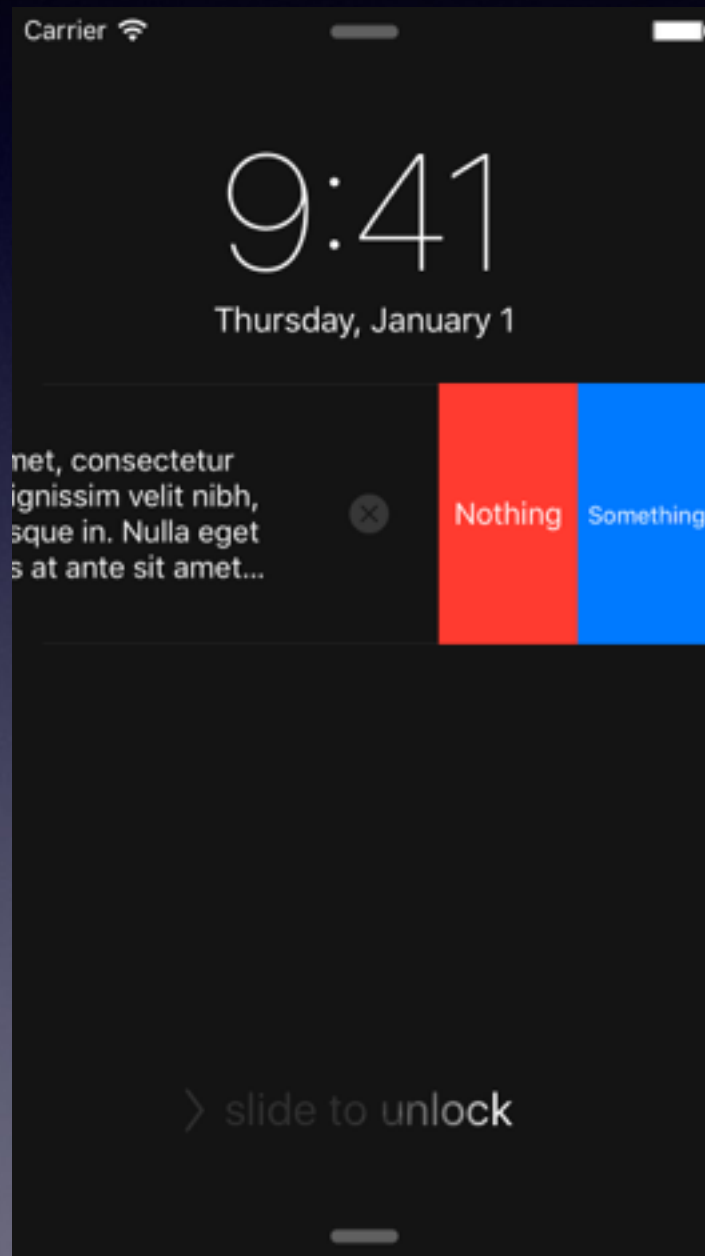


# Categories & Actions

```
let action = UIMutableUserNotificationAction()  
action.identifier = "action-identifier"  
action.title = "Action"  
action.activationMode = UIUserNotificationActivationMode.Foreground  
action.authenticationRequired = true  
action.destructive = true
```



# Categories & Actions



# Categories & Actions

```
let action = UIMutableUserNotificationAction()  
action.identifier = "action-identifier"  
action.title = "Action"  
action.activationMode = UIUserNotificationActivationMode.Background  
action.authenticationRequired = false  
action.destructive = false
```

# Categories & Actions



# Categories & Actions



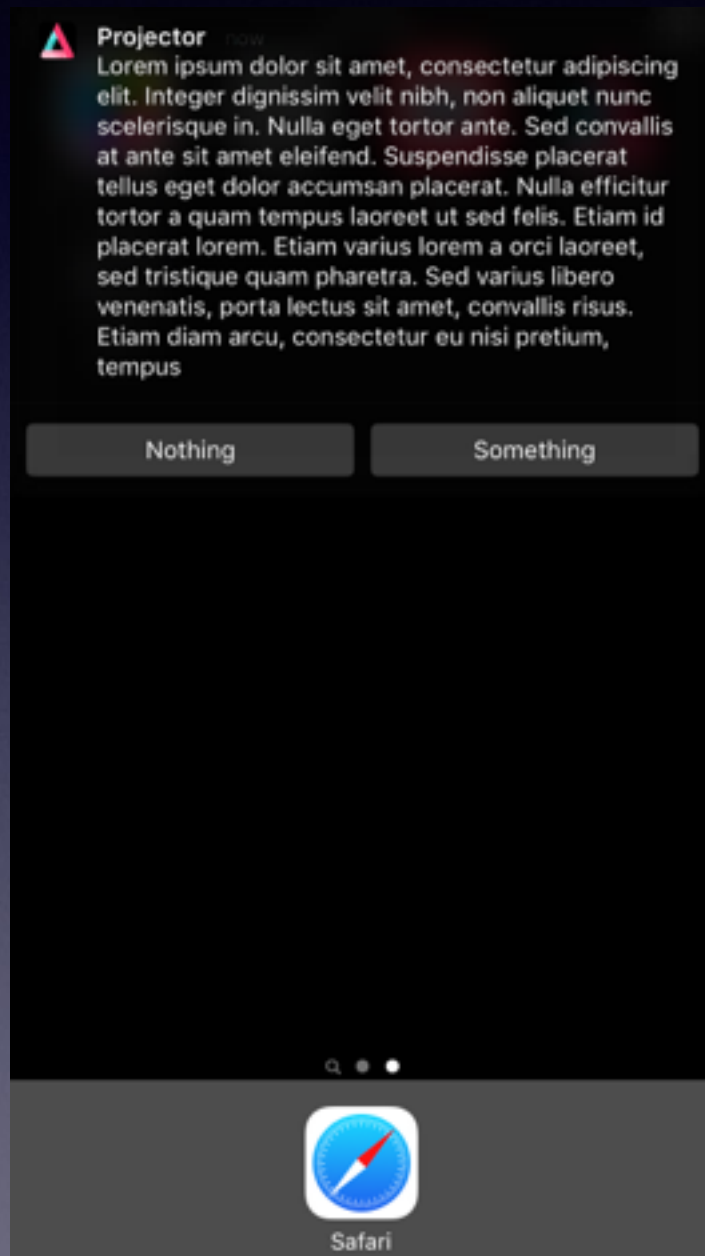
# Categories & Actions



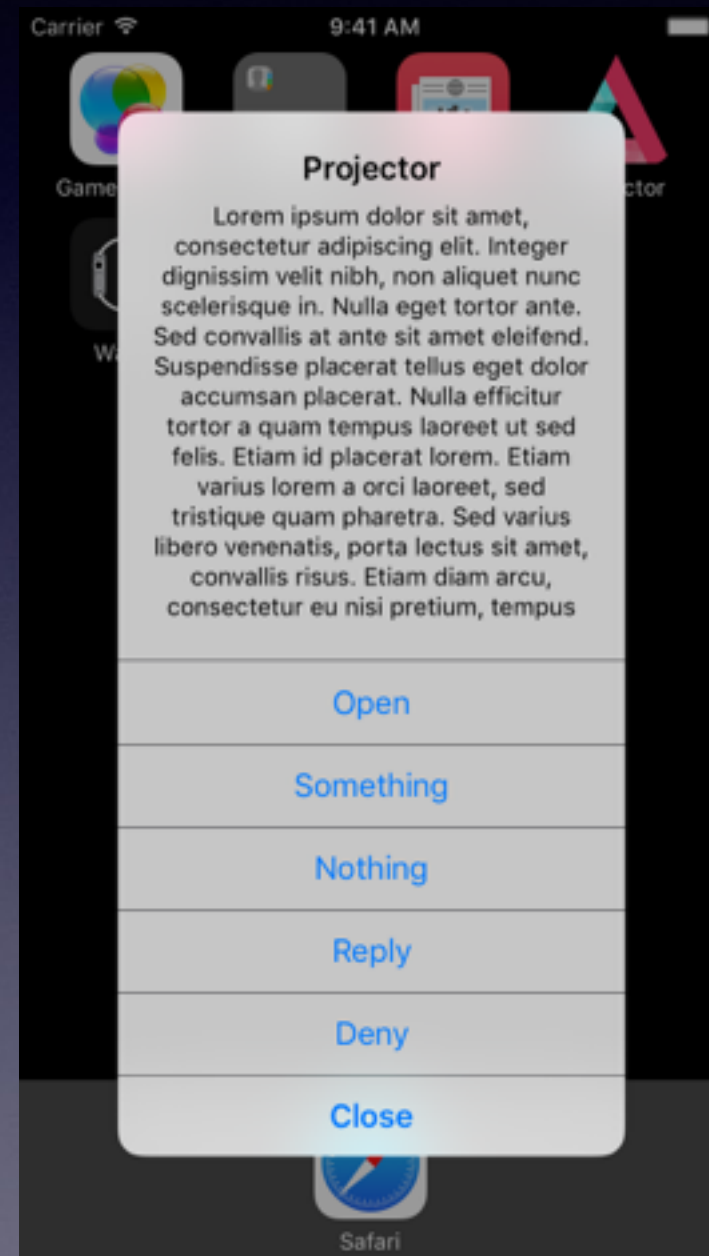
```
let category = NSMutableUserNotificationCategory()  
category.identifier = "category-identifier"  
category.setActions(actions, forContext:  
    UIUserNotificationActionContext.Default)  
category.setActions(actions, forContext:  
    UIUserNotificationActionContext.Minimal)
```



# Categories & Actions



.Minimal



.Default

# Categories & Actions

```
let notificationTypes: UIUserNotificationType =  
    [.Alert, .Badge, .Sound]
```

```
let settings = UIUserNotificationSettings(  
    forTypes: notificationType,  
    categories: [category])
```

```
UIApplication.sharedApplication()  
    .registerUserNotificationSettings(settings)
```

# What's Changed in iOS 10?

# What's Changed in iOS 10?

- New `UserNotifications.framework`



# What's Changed in iOS 10?

- New `UserNotifications.framework`
- All Notifications are now `UNNotification`



# What's Changed in iOS 10?

- New `UserNotifications.framework`
- All Notifications are now `UNNotification`
- (Almost) Everything Handled via `UNUserNotificationCenter`

# What's Changed in iOS 10?

- New `UserNotifications.framework`
- All Notifications are now `UNNotification`
- (Almost) Everything Handled via `UNUserNotificationCenter`
- Much simpler delegation callbacks

# What's Changed in iOS 10?

UIApplication

- .registerUserNotificationSettings(\_:)
- .registerForRemoteNotifications()

UIApplicationDelegate

- .application(\_:didRegisterUserNotificationSettings:)
- .application(\_:didRegisterForRemoteNotificationsWithDeviceToken:)
- .application(\_:didFailToRegisterForRemoteNotificationsWithError:)

# What's Changed in iOS 10?

UIApplication

~~——.registerUserNotificationSettings(\_:)~~  
    .registerForRemoteNotifications()

UIApplicationDelegate

~~——.application(\_:didRegisterUserNotificationSettings:)~~  
    .application(\_:didRegisterForRemoteNotificationsWithDeviceToken:)  
    .application(\_:didFailToRegisterForRemoteNotificationsWithError:)

UNUserNotificationCenter

    .requestAuthorization(options:completionHandler:)  
    .setNotificationCategories(\_:)



# What's Changed in iOS 10?

UIApplication

~~——.registerUserNotificationSettings(\_:)~~  
    .registerForRemoteNotifications()

UIApplicationDelegate

~~——.application(\_:didRegisterUserNotificationSettings:)~~  
    .application(\_:didRegisterForRemoteNotificationsWithDeviceToken:)  
    .application(\_:didFailToRegisterForRemoteNotificationsWithError:)

UNUserNotificationCenter

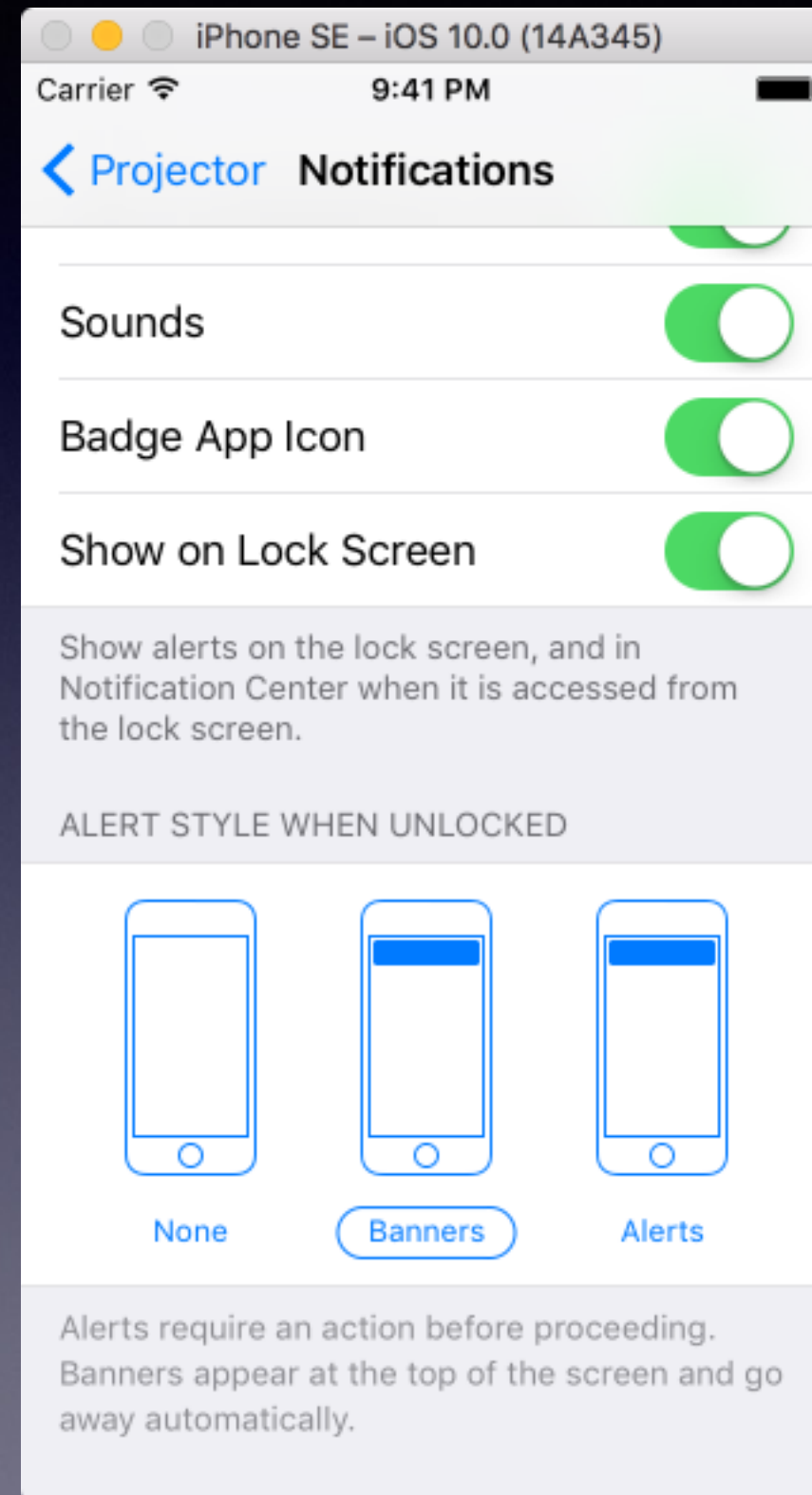
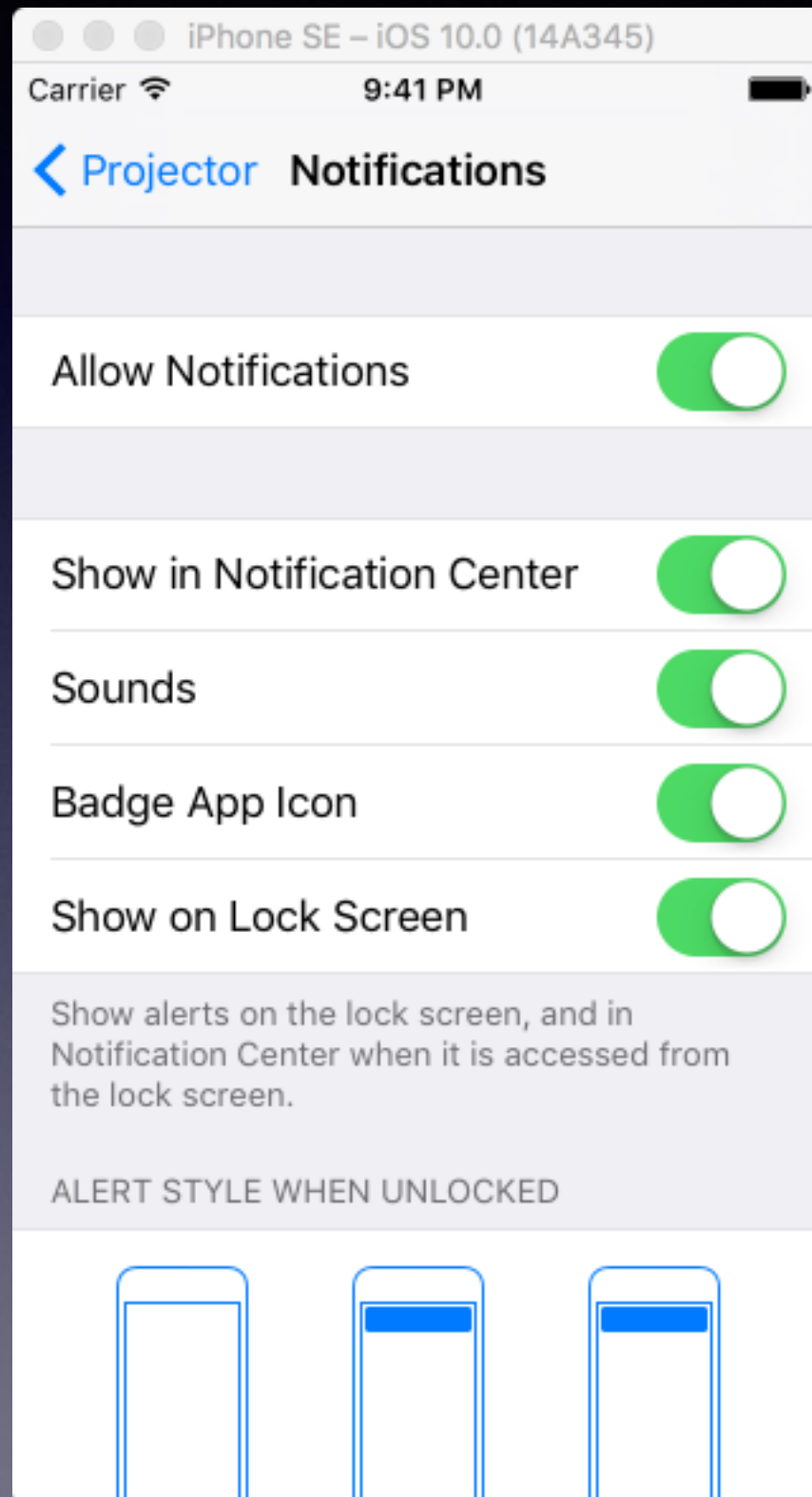
    .requestAuthorization(options:completionHandler:)  
    .setNotificationCategories(\_:)

UNUserNotificationCenter

    .getNotificationSettings(completionHandler:)  
    .getNotificationCategories(completionHandler:)



# What's Changed in iOS 10?



# What's Changed in iOS 10?

UIApplication

- .scheduleLocalNotification(\_:)
- .presentLocalNotificationNow(\_:)
- .cancelLocalNotification(\_:)
- .cancelAllLocalNotifications()
- .scheduledLocalNotifications()

# What's Changed in iOS 10?

## ~~UIApplication~~

~~——.scheduleLocalNotification(\_:)~~  
~~——.presentLocalNotificationNow(\_:)~~  
~~——.cancelLocalNotification(\_:)~~  
~~——.cancelAllLocalNotifications()~~  
~~——.scheduledLocalNotifications()~~

## UNUserNotificationCenter

.add(\_:withCompletionHandler:)  
.getPendingNotificationRequests(completionHandler:)  
.removePendingNotificationRequests(withIdentifiers:)  
.removeAllPendingNotificationRequests()

## UNUserNotificationCenter

.getDeliveredNotifications(completionHandler:)  
.removeDeliveredNotifications(withIdentifiers:)  
.removeAllDeliveredNotifications()

# What's Changed in iOS 10?

```
UNUserNotificationCenter  
    .add(_ request: UNNotificationRequest  
        completionHandler completion: ((Error?) -> Void)?)
```

```
class UNNotificationRequest {  
    var identifier: String { get }  
    var content: UNNotificationContent { get }  
    var trigger: UNNotificationTrigger? { get }  
}
```



# What's Changed in iOS 10?

```
class UN[Mutable]NotificationContent {  
    var title: String  
    var subtitle: String  
    var body: String  
    var badge: NSNumber?  
    var sound: UNNotificationSound?  
    var userInfo: [AnyHashable : Any]  
    var launchImageName: String  
    var categoryIdfier: String  
  
}
```



# What's Changed in iOS 10?

```
class UN[Mutable]NotificationContent {  
    var title: String  
    var subtitle: String  
    var body: String  
    var badge: NSNumber?  
    var sound: UNNotificationSound?  
    var userInfo: [AnyHashable : Any]  
    var launchImageName: String  
    var categoryIdentifier: String  
  
    var threadIdentifier: String  
    var attachments: [UNNotificationAttachment]  
}
```

# What's Changed in iOS 10?

```
class UNNotificationTrigger {  
    open var repeats: Bool { get }  
}
```

```
class UNPushNotificationTrigger : UNNotificationTrigger { }
```

```
class UNTimeIntervalNotificationTrigger : UNNotificationTrigger {  
    var timeInterval: TimeInterval { get }  
    init(timeInterval: TimeInterval, repeats: Bool)  
    func nextTriggerDate() -> Date?  
}
```

```
class UNCalendarNotificationTrigger : UNNotificationTrigger {  
    var dateComponents: DateComponents { get }  
    init(dateMatching dateComponents: DateComponents, repeats: Bool)  
    func nextTriggerDate() -> Date?  
}
```

```
class UNLocationNotificationTrigger : UNNotificationTrigger {  
    var region: CLRegion { get }  
    init(region: CLRegion, repeats: Bool)  
}
```

# What's Changed in iOS 10?

## UNUserNotificationCenterDelegate

```
protocol UNUserNotificationCenterDelegate {  
  
    func userNotificationCenter(  
        _ center: UNUserNotificationCenter,  
        willPresent notification: UNNotification,  
        withCompletionHandler  
            completionHandler: (UNNotificationPresentationOptions) -> Void)  
  
    func userNotificationCenter(  
        _ center: UNUserNotificationCenter,  
        didReceive response: UNNotificationResponse,  
        withCompletionHandler completionHandler: () -> Void)  
  
}
```

# Categories & Actions

```
UNNotificationCenter.setNotificationCategories(_:)
```

```
class UNNotificationCategory {  
    var identifier: String { get }  
    var actions: [UNNotificationAction] { get }  
    var intentIdentifiers: [String] { get }  
    var options: UNNotificationCategoryOptions { get }  
    init(identifier: String,  
        actions: [UNNotificationAction],  
        intentIdentifiers: [String],  
        options: UNNotificationCategoryOptions = [])  
}
```



# Categories & Actions

```
UNNotificationCenter.setNotificationCategories(_:)
```

```
class UNNotificationCategory {  
    var identifier: String { get }  
    var actions: [UNNotificationAction] { get }  
    var intentIdentifiers: [String] { get }  
    var options: UNNotificationCategoryOptions { get }  
    init(identifier: String,  
          actions: [UNNotificationAction],  
          intentIdentifiers: [String],  
          options: UNNotificationCategoryOptions = [])  
}
```

```
struct UNNotificationCategoryOptions : OptionSet {  
    static var customDismissAction: UNNotificationCategoryOptions {get}  
    static var allowInCarPlay: UNNotificationCategoryOptions { get }  
}
```



# Categories & Actions

```
UNNotificationCenter.setNotificationCategories(_:)
```

```
class UNNotificationAction {  
    var identifier: String { get }  
    var title: String { get }  
    var options: UNNotificationActionOptions { get }  
    init(identifier: String,  
         title: String,  
         options: UNNotificationActionOptions = [])  
}
```

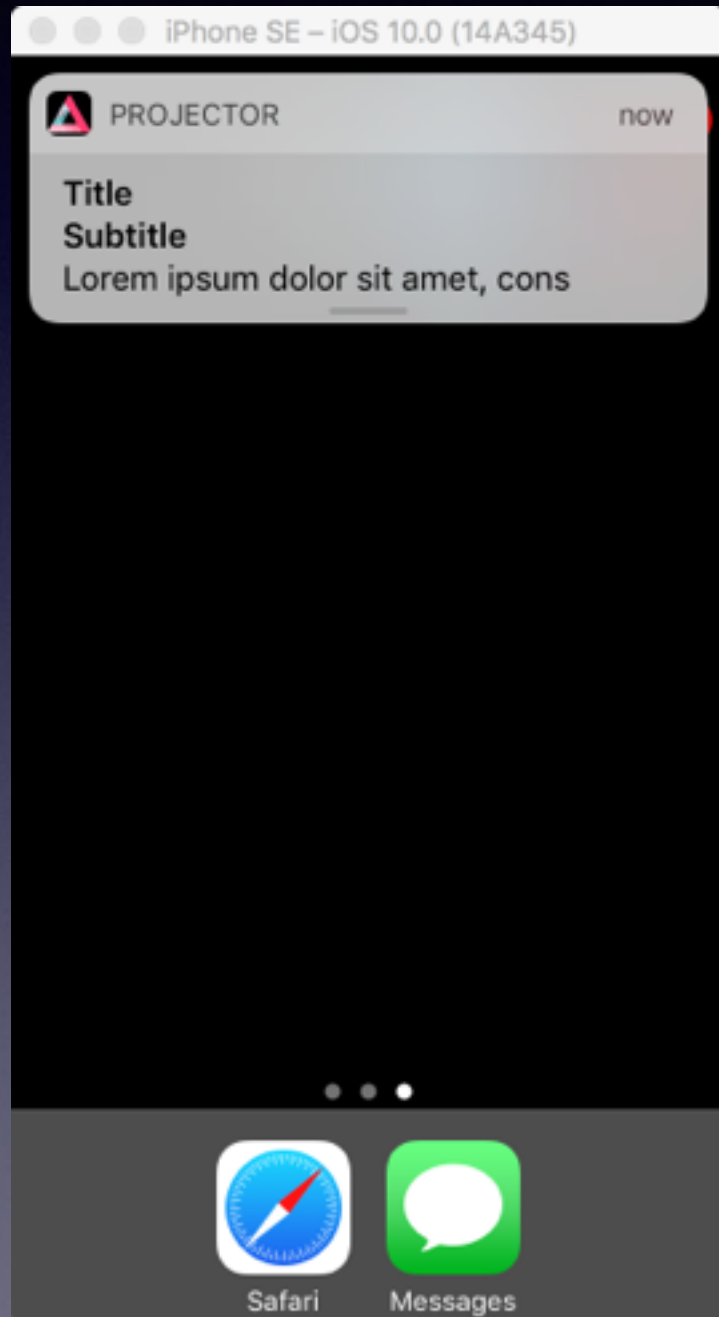
# Categories & Actions

```
UNNotificationCenter.setNotificationCategories(_:)
```

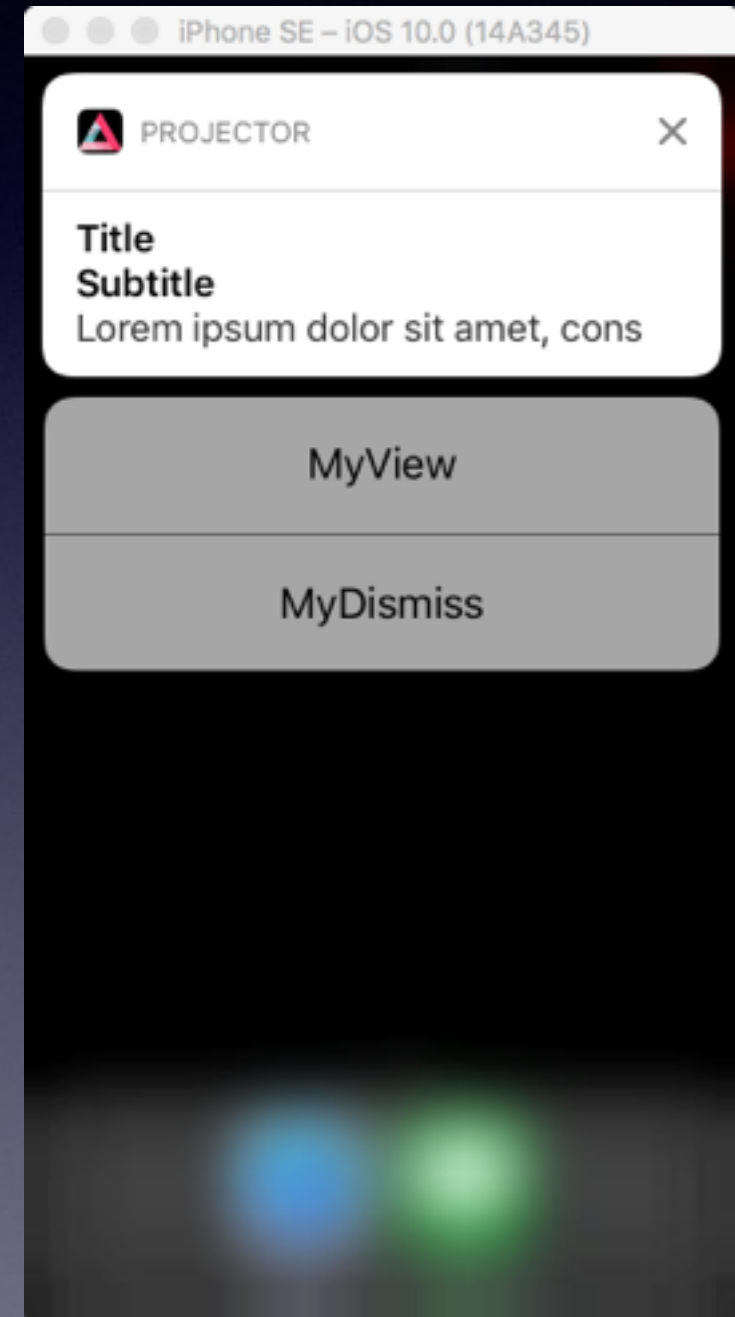
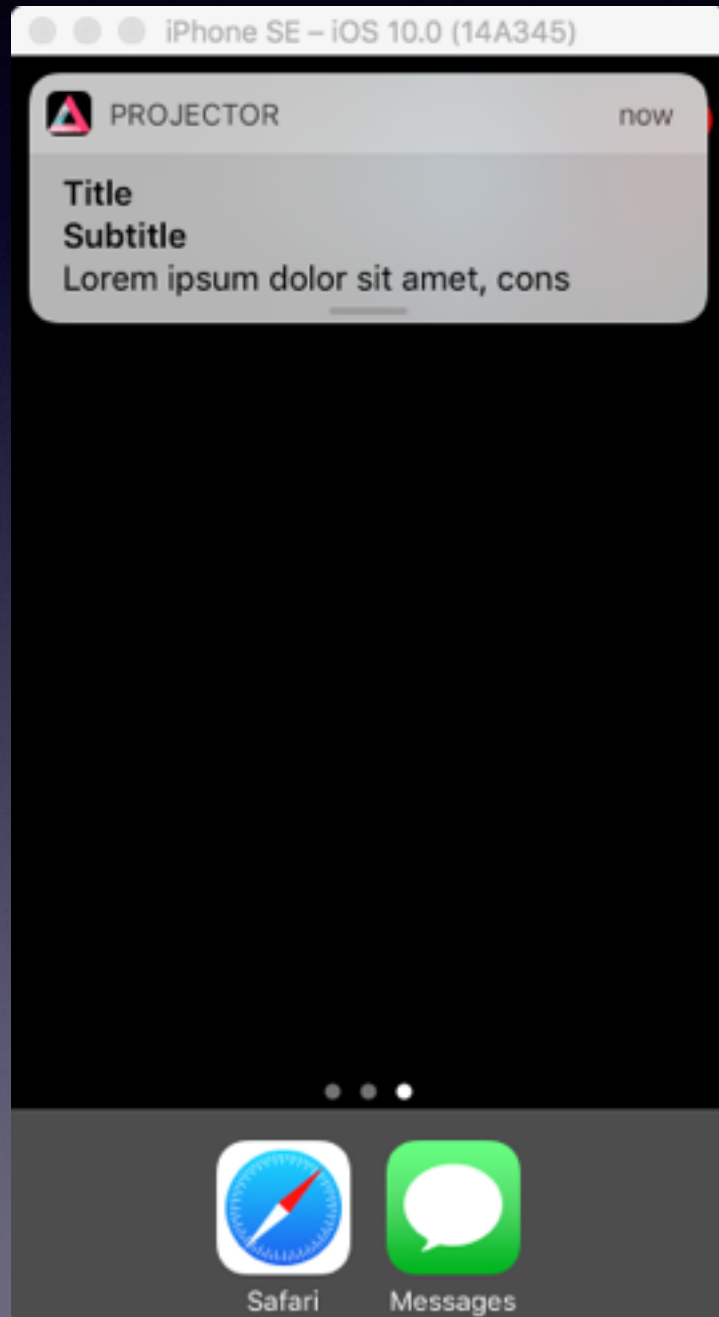
```
class UNNotificationAction {  
    var identifier: String { get }  
    var title: String { get }  
    var options: UNNotificationActionOptions { get }  
    init(identifier: String,  
         title: String,  
         options: UNNotificationActionOptions = [])  
}
```

```
struct UNNotificationActionOptions {  
    static var authenticationRequired: UNNotificationActionOptions {get}  
    static var destructive: UNNotificationActionOptions { get }  
    static var foreground: UNNotificationActionOptions { get }  
}
```

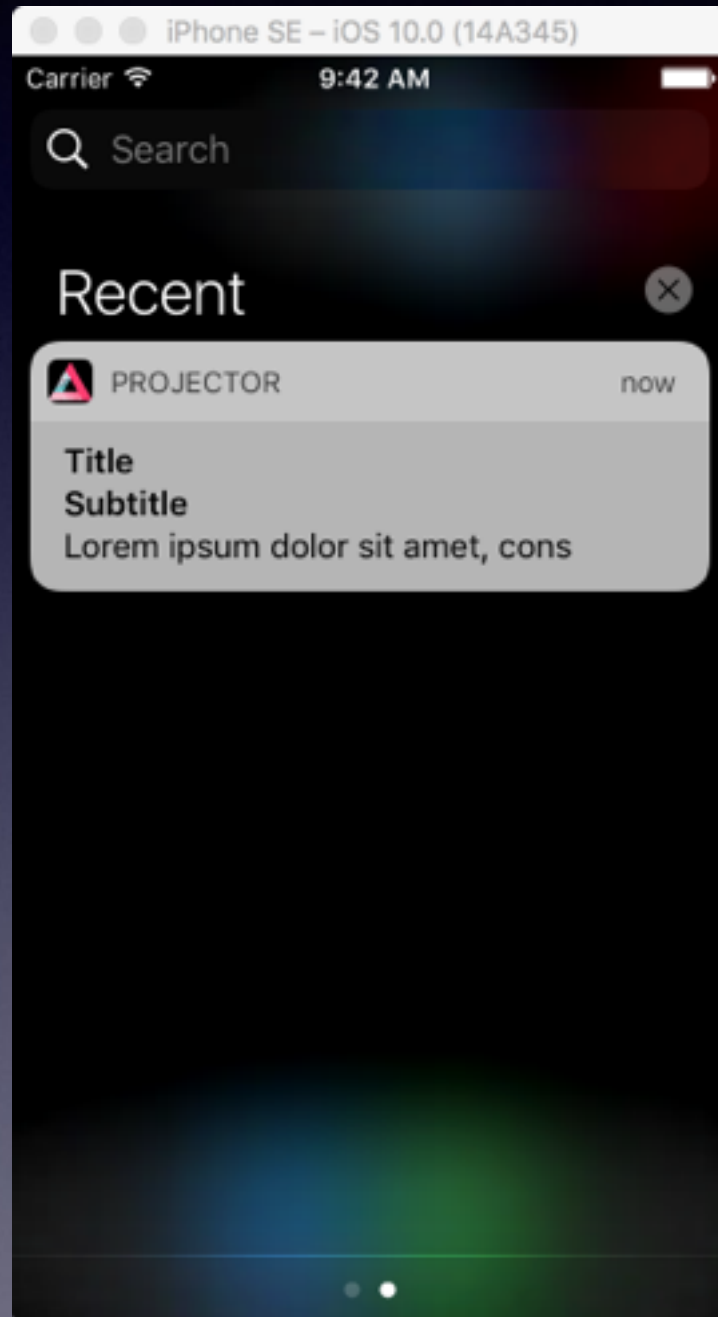
# Categories & Actions



# Categories & Actions

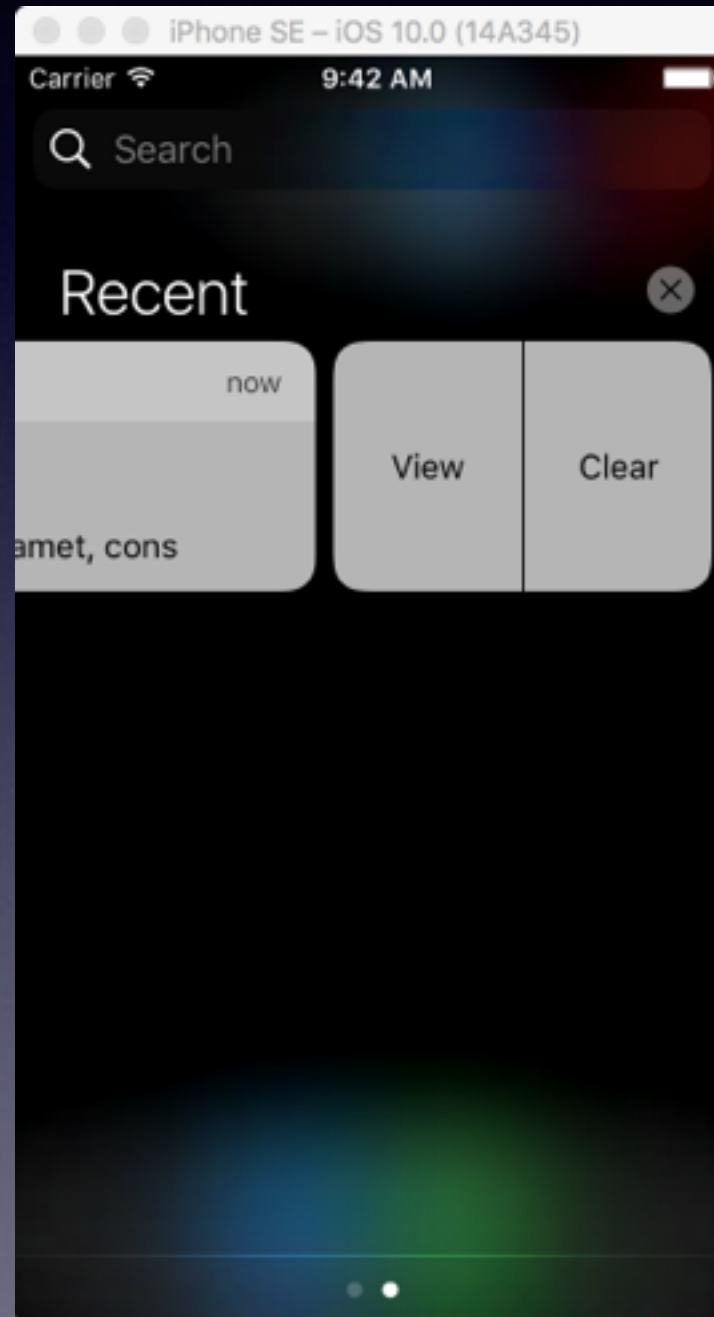
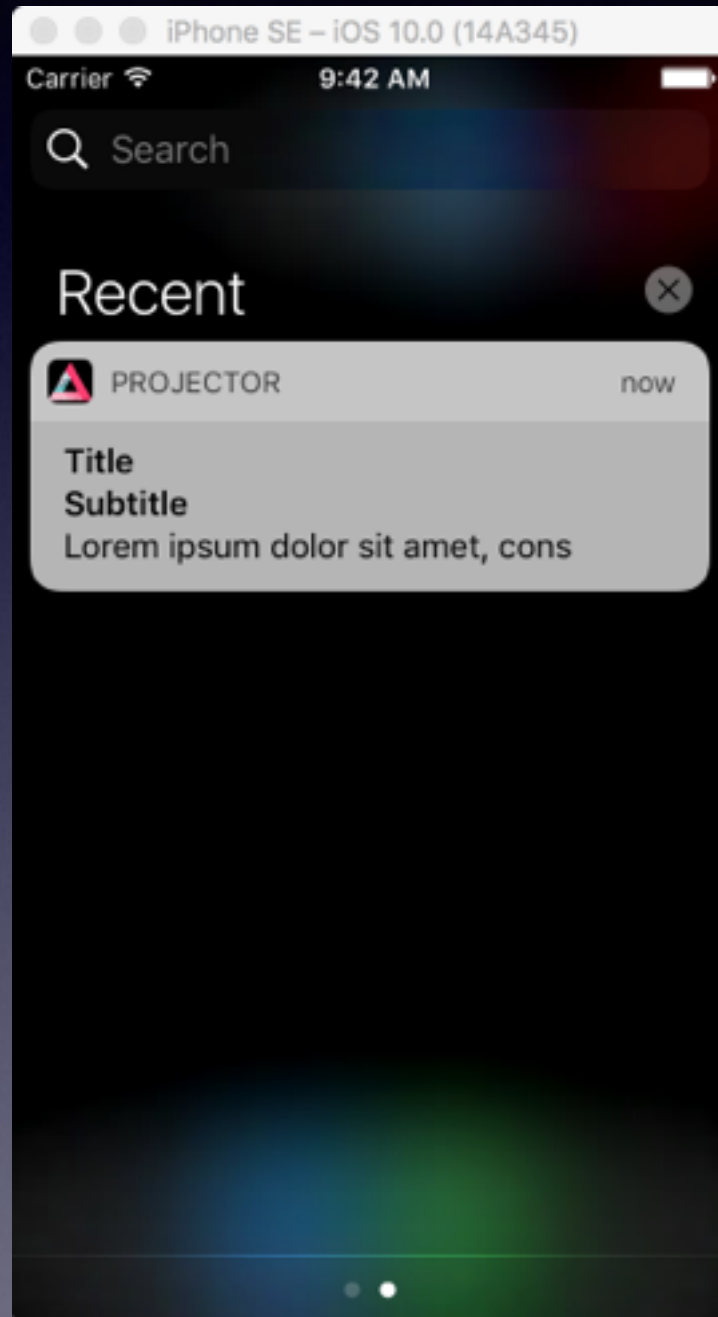


# Categories & Actions

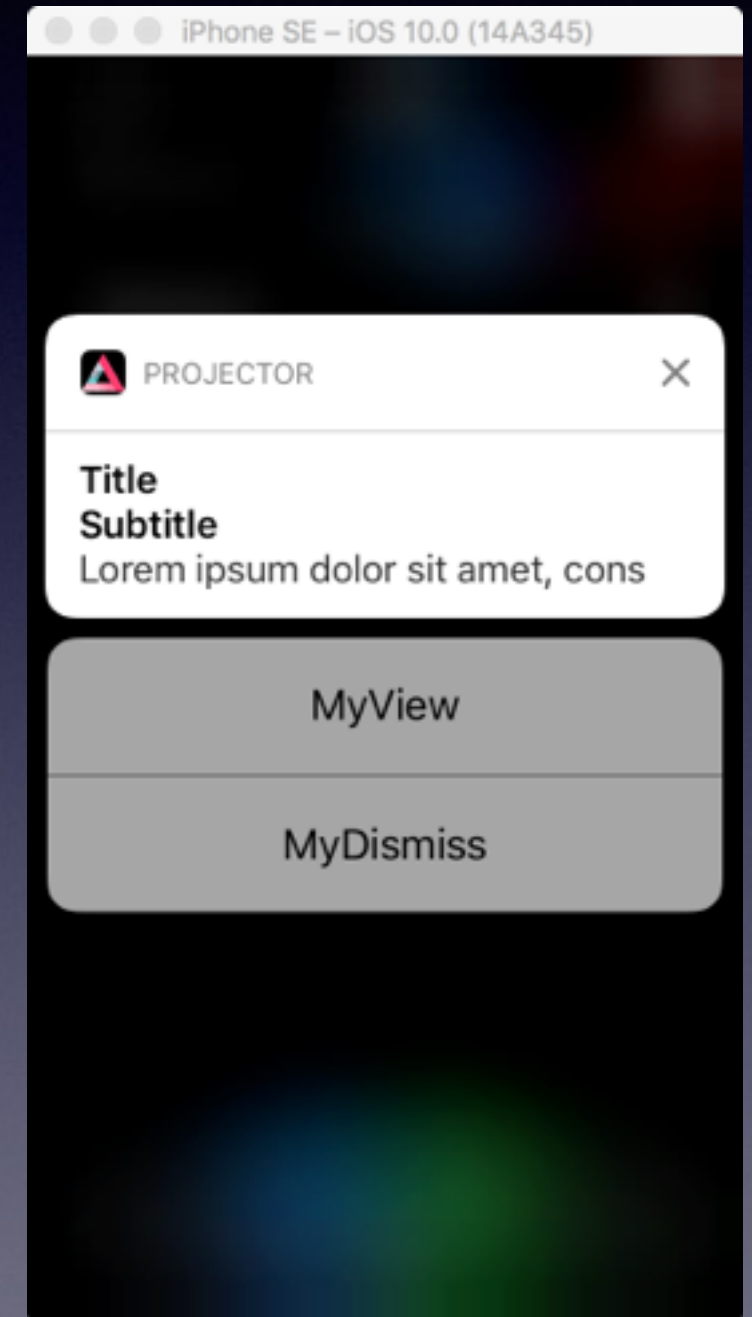
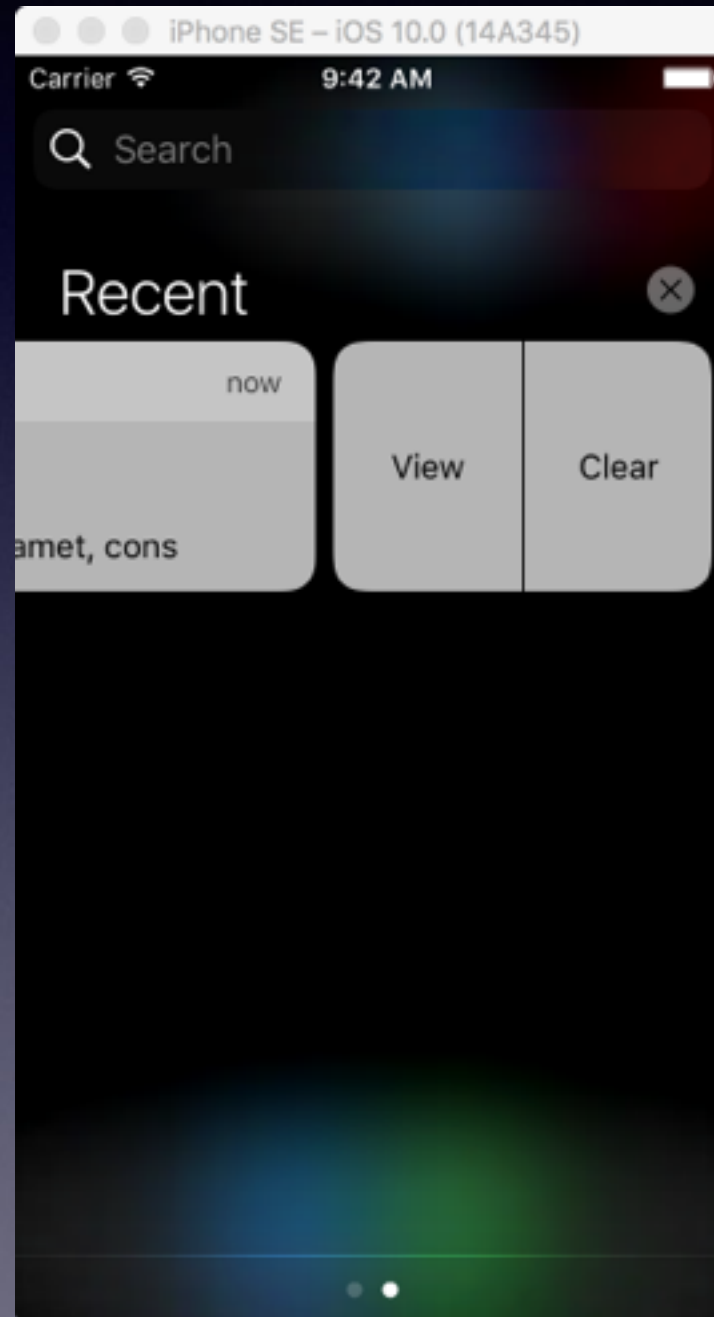
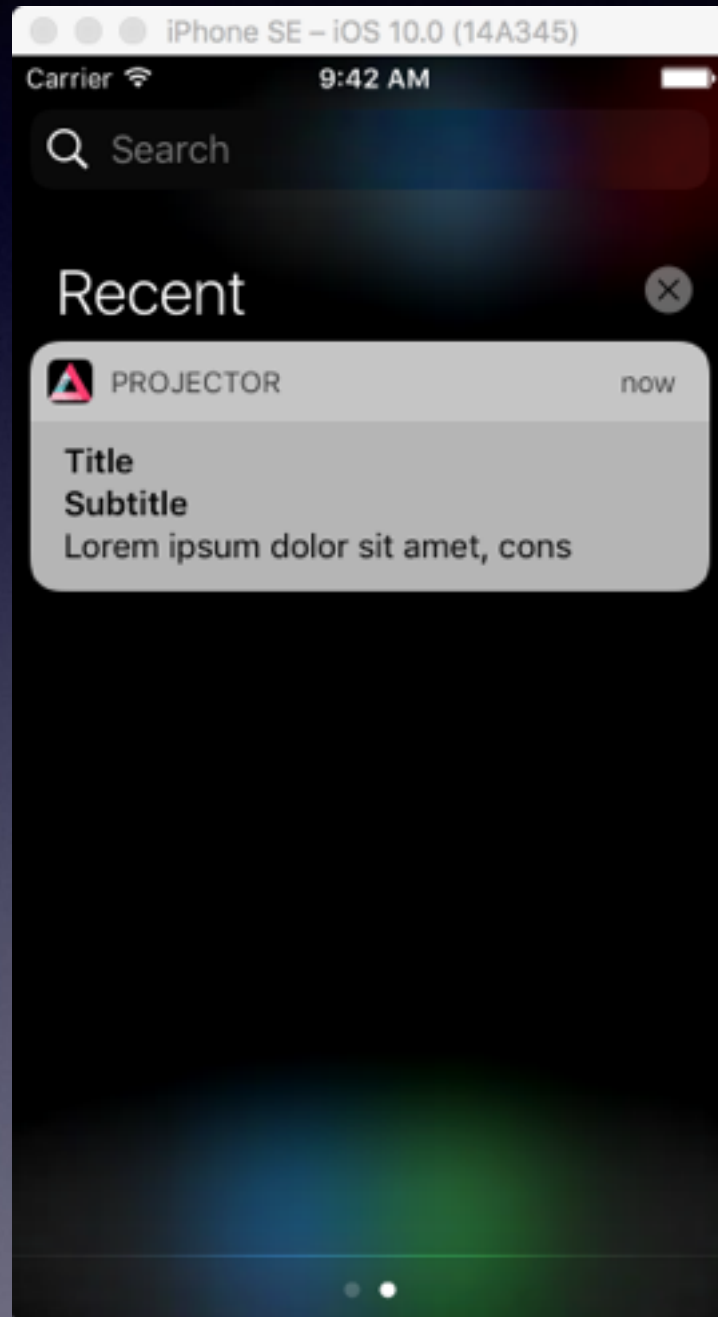




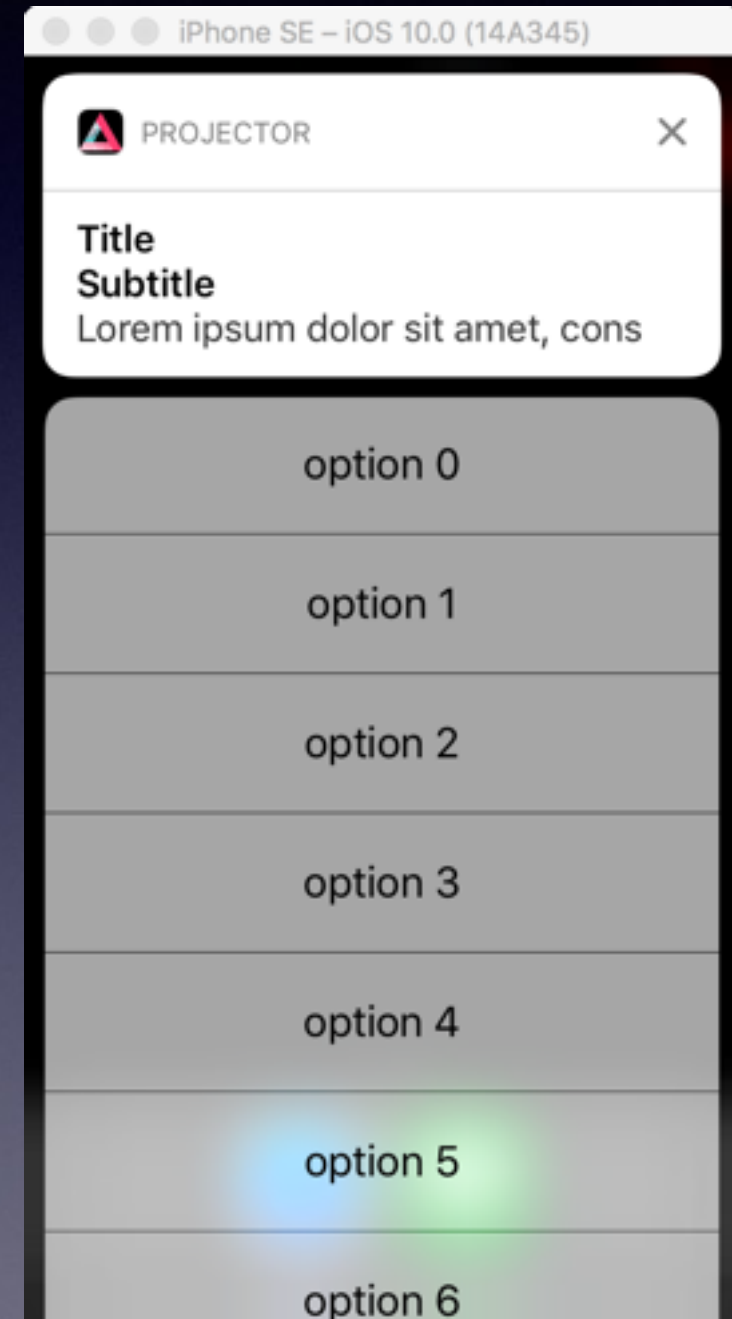
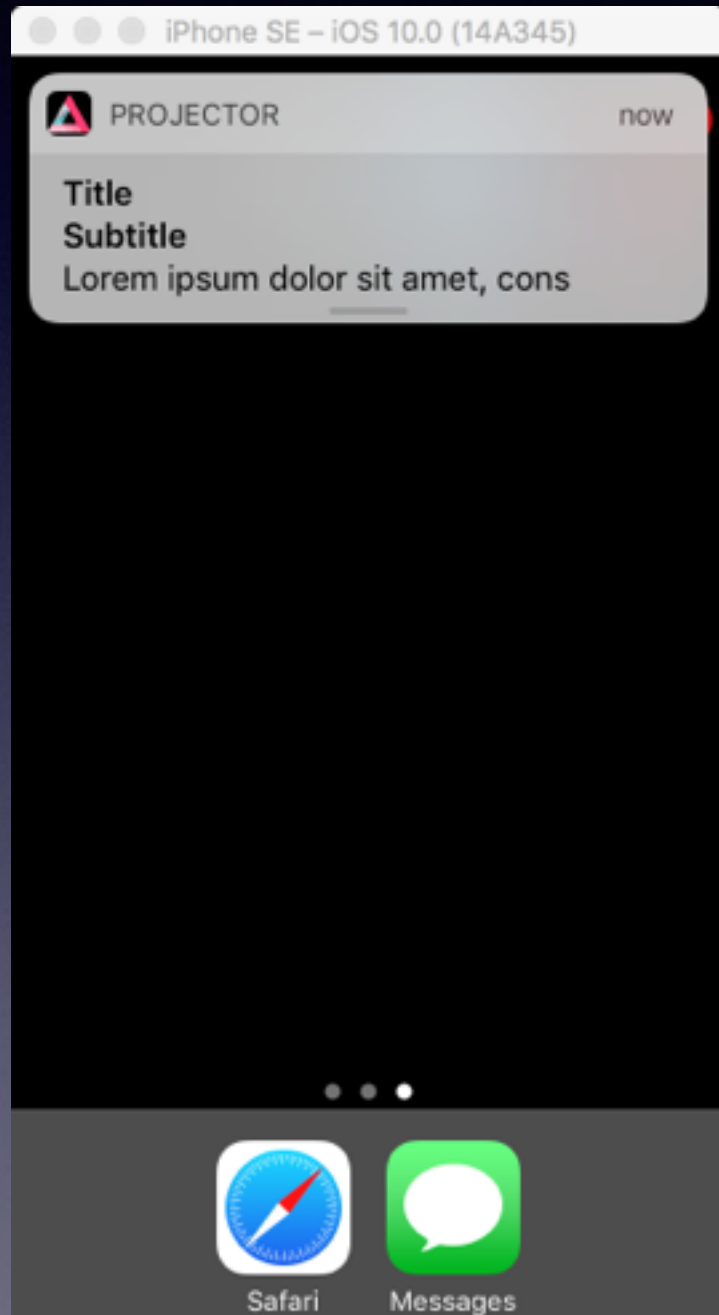
# Categories & Actions



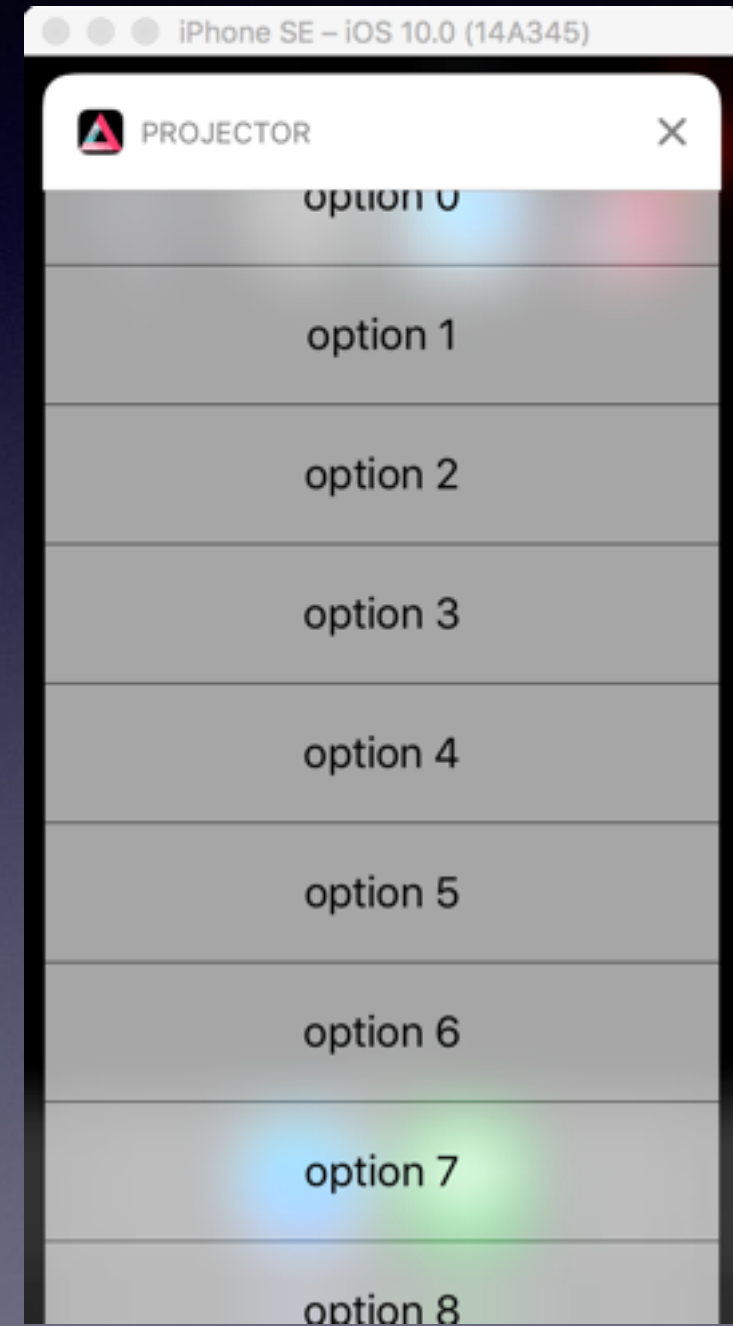
# Categories & Actions



# Categories & Actions



# Categories & Actions

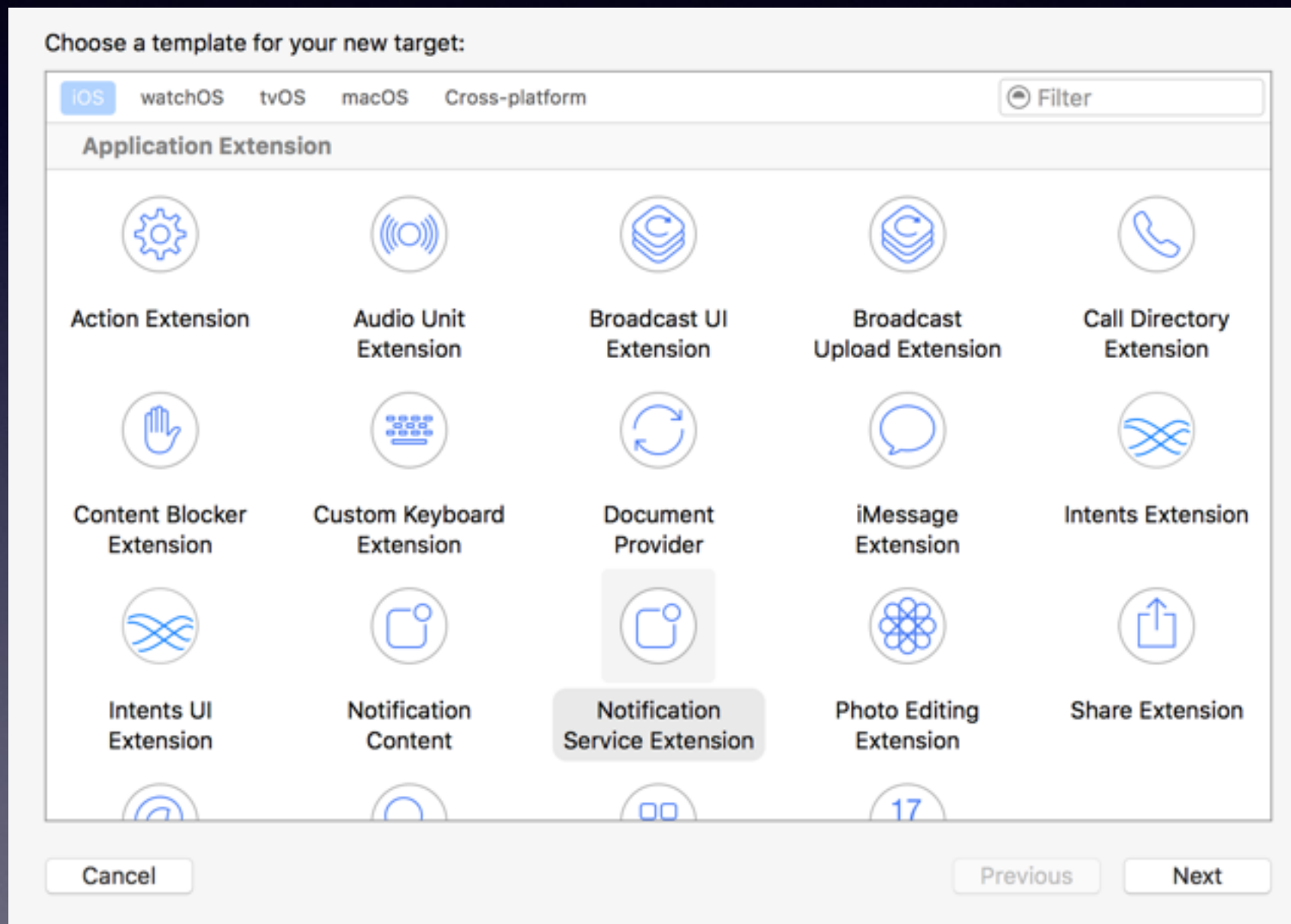


# Service Extension

- Intercept Remote Notification
- Add `mutable-content: 1` to APNS payload
- Limited Time to Process (~30 seconds wall clock time)
- Basically an App Extension (with all limitations)



# Service Extension



# Service Extension

```
class NotificationService: UNNotificationServiceExtension {  
    var contentHandler: ((UNNotificationContent) -> Void)?  
    var bestAttemptContent: UNMutableNotificationContent?  
  
    override func didReceive(_ request: UNNotificationRequest,  
                             withContentHandler contentHandler: @escaping (UNNotificationContent) -> Void) {  
        self.contentHandler = contentHandler  
        bestAttemptContent = (request.content.mutableCopy() as? UNMutableNotificationContent)  
  
        if let bestAttemptContent = bestAttemptContent {  
            bestAttemptContent.title = "\(bestAttemptContent.title) [modified]"  
  
            contentHandler(bestAttemptContent)  
        }  
    }  
  
    override func serviceExtensionTimeWillExpire() {  
        if let contentHandler = contentHandler, let bestAttemptContent = bestAttemptContent {  
            contentHandler(bestAttemptContent)  
        }  
    }  
}
```

# Service Extension

```
override func didReceive(_ request: UNNotificationRequest,
                        withContentHandler contentHandler:
                            @escaping (UNNotificationContent) -> Void) {
    self.contentHandler = contentHandler

    bestAttemptContent =
        (request.content.mutableCopy() as? UNMutableNotificationContent)

    if let bestAttemptContent = bestAttemptContent {
        bestAttemptContent.title = "\(bestAttemptContent.title) [modified]"

        contentHandler(bestAttemptContent)
    }
}
```

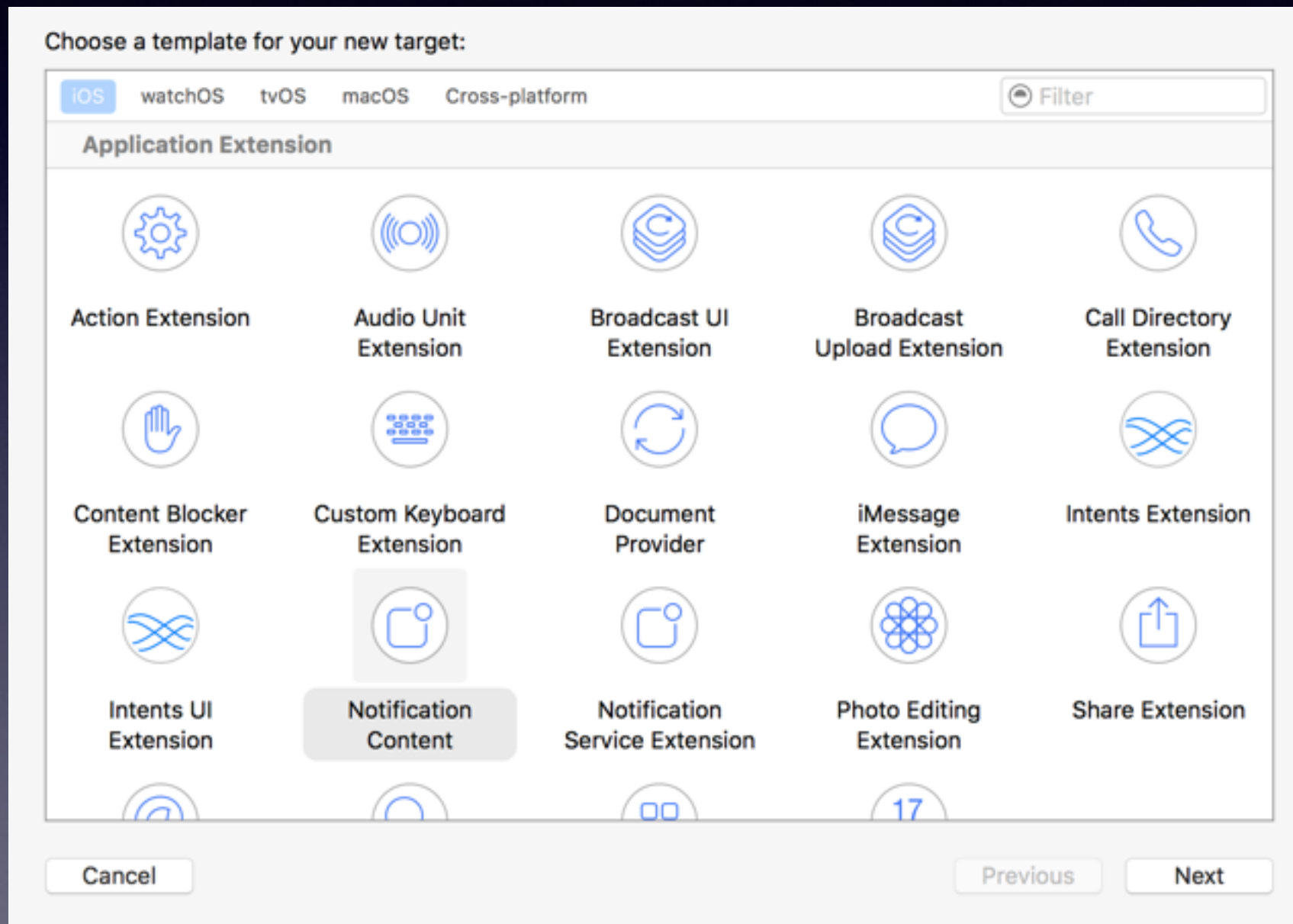
# Service Extension

```
override func serviceExtensionTimeWillExpire() {  
    if let contentHandler = contentHandler,  
        let bestAttemptContent = bestAttemptContent {  
        contentHandler(bestAttemptContent)  
    }  
}
```



# Content Extension

# Content Extension



# Content Extension

```
class NotificationViewController: UIViewController,
UNNotificationContentExtension {

    @IBOutlet var label: UILabel?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any required interface initialization here.
    }

    func didReceive(_ notification: UNNotification) {
        self.label?.text = notification.request.content.body
    }

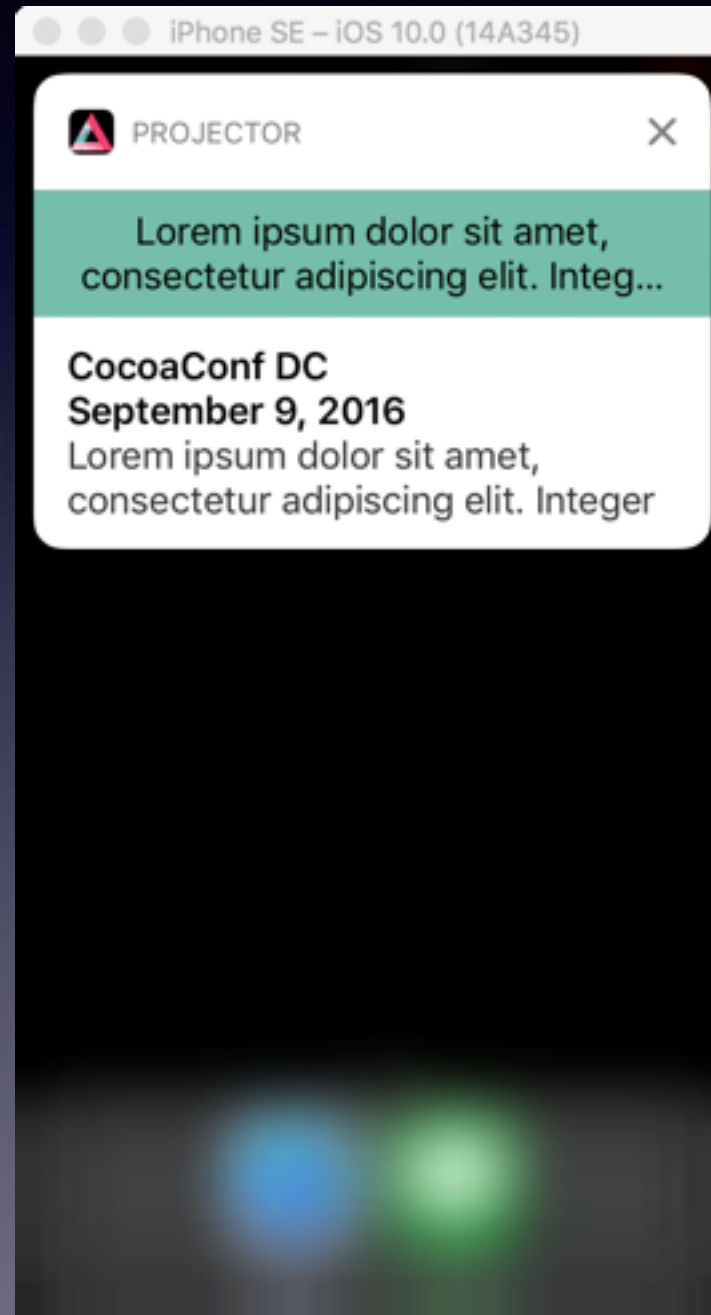
}
```

# Content Extension

Key	Type	Value
Information Property List	Dictionary	(10 items)
Localization native development region	String	en
Bundle display name	String	NotificationContent
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
NSExtension	Dictionary	(3 items)
NSExtensionAttributes	Dictionary	(2 items)
UNNotificationExtensionCategory	String	myNotificationCategory
UNNotificationExtensionInitialContentSizeRatio	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension



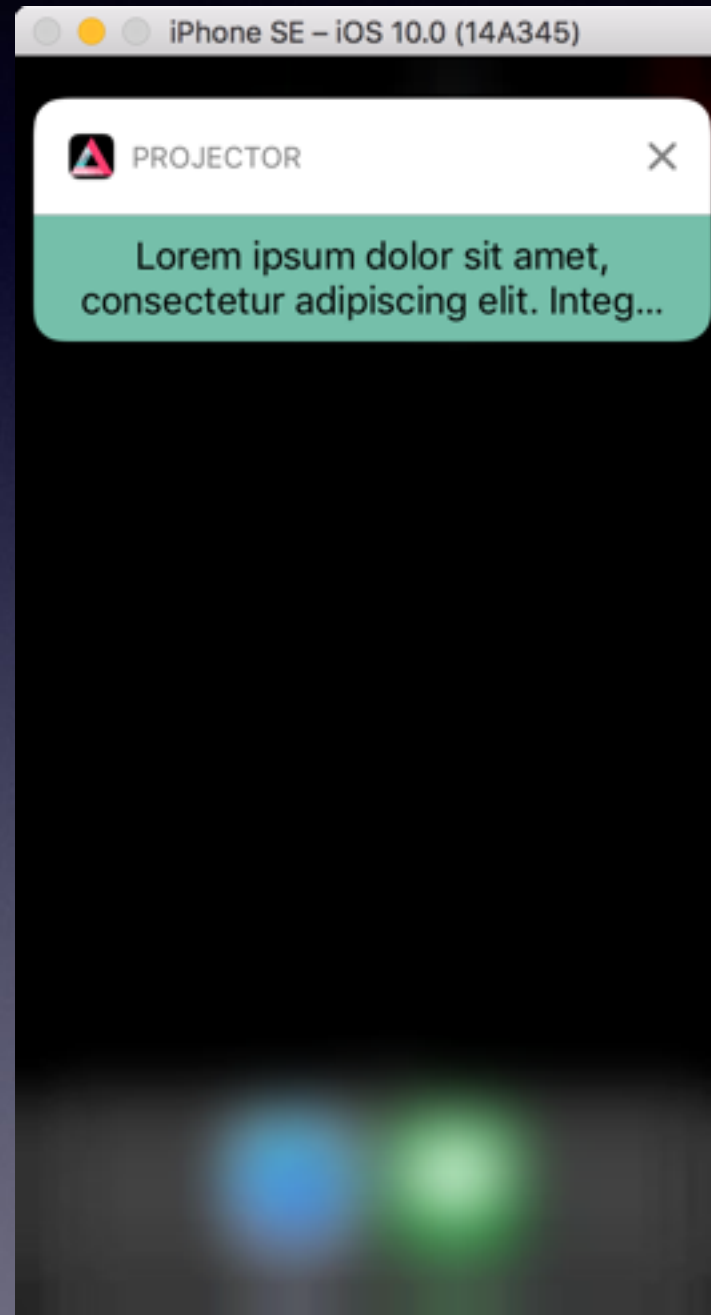
# Content Extension



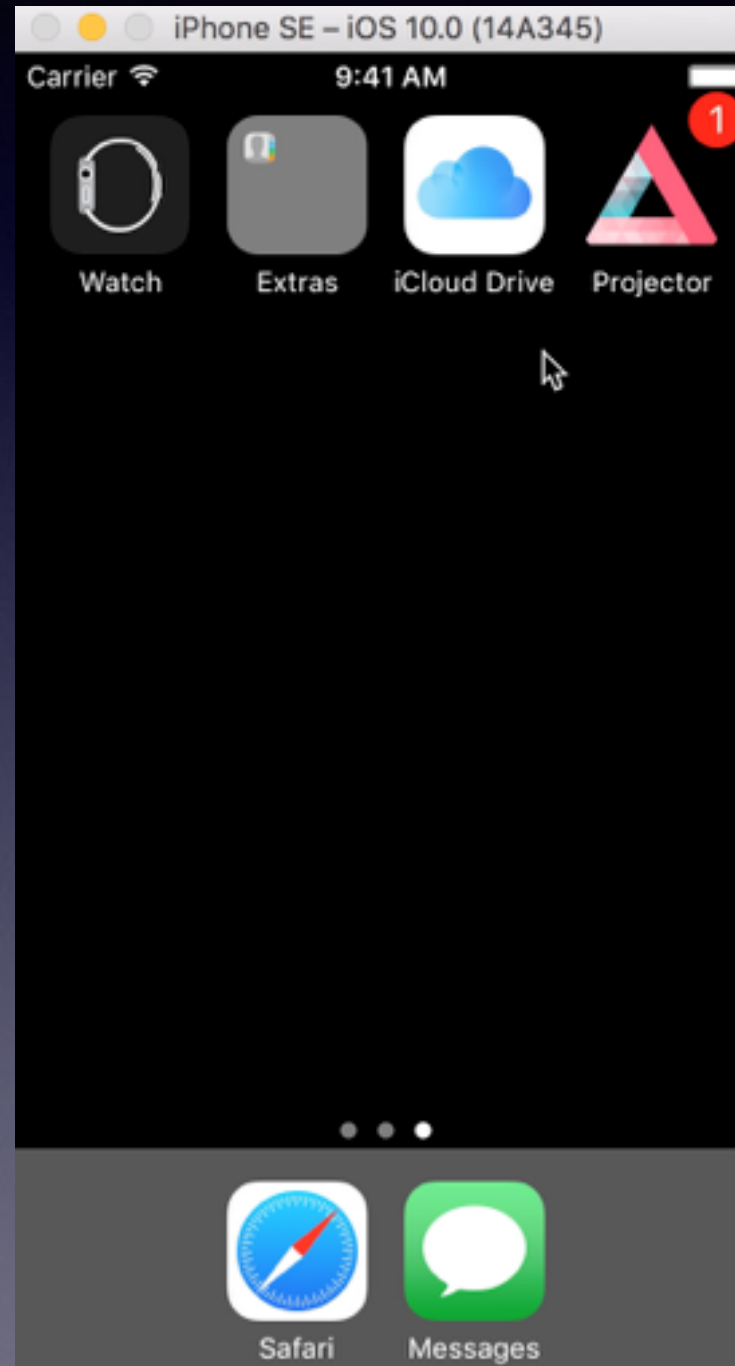
# Content Extension

Key	Type	Value
▼ Information Property List	Dictionary	(10 items)
Localization native development region	String	en
Bundle display name	String	NotificationContent
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	XPC!
Bundle versions string, short	String	1.0
Bundle version	String	1
▼ NSExtension	Dictionary	(3 items)
▼ NSExtensionAttributes	Dictionary	(3 items)
UNNotificationExtensionCategory	String	NotificationContentCategory
UNNotificationExtensionDefaultContentHidden	Boolean	YES
UNNotificationExtensionInitialContentSizeRatio	Number	1
NSExtensionMainStoryboard	String	MainInterface
NSExtensionPointIdentifier	String	com.apple.usernotifications.content-extension

# Content Extension



# Content Extension

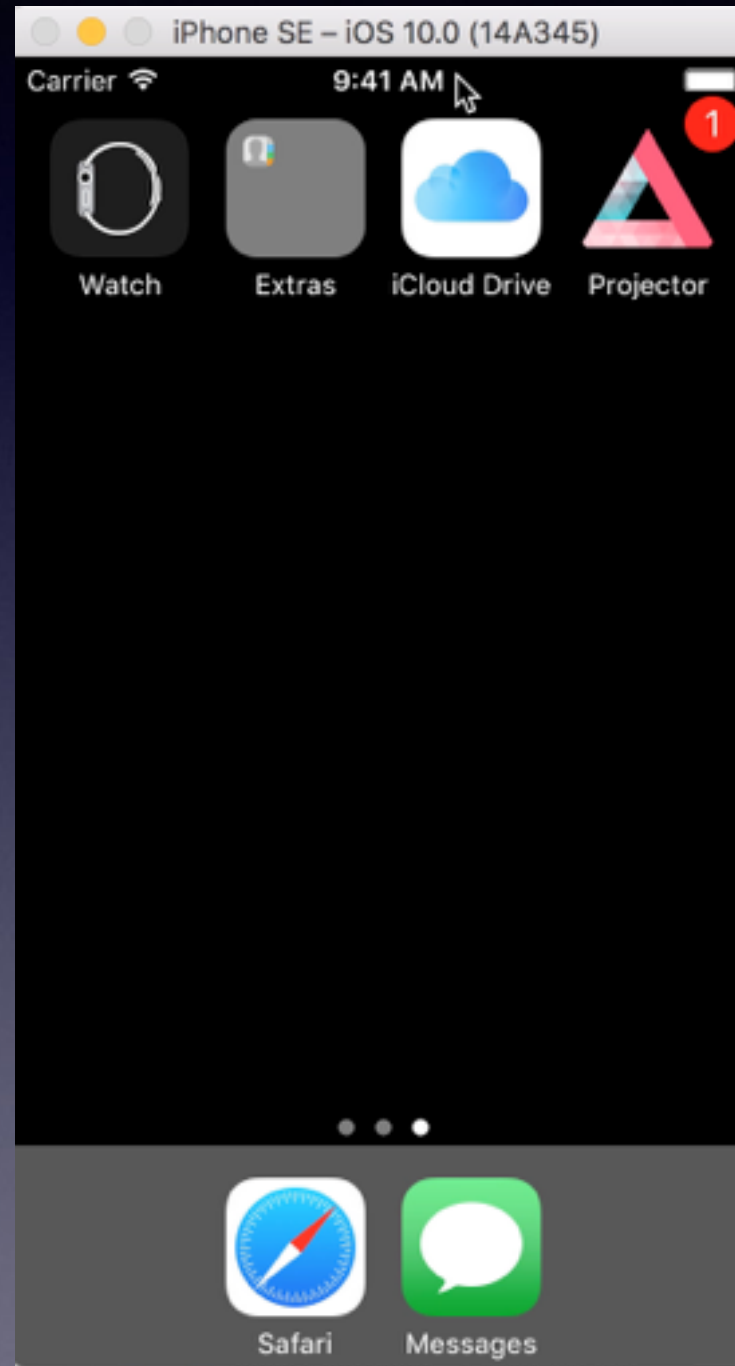




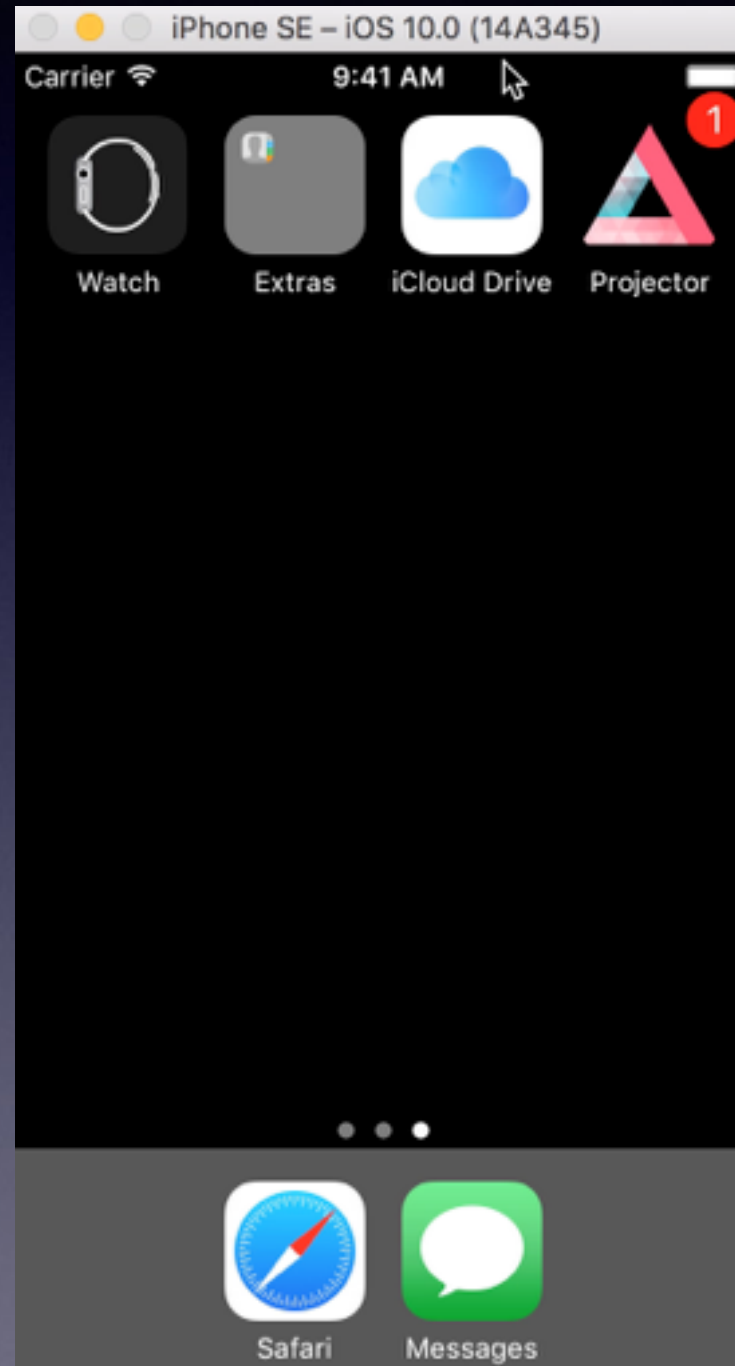
# Content Extension

Key		Type	Value
▼ Information Property List		Dictionary	(10 items)
Localization native development region	↕	String	en
Bundle display name	↕	String	NotificationContent
Executable file	↕	String	\$(EXECUTABLE_NAME)
Bundle identifier	↕	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	↕	String	6.0
Bundle name	↕	String	\$(PRODUCT_NAME)
Bundle OS Type code	↕	String	XPC!
Bundle versions string, short	↕	String	1.0
Bundle version	↕	String	1
▼ NSExtension	↕	Dictionary	(3 items)
▼ NSExtensionAttributes		Dictionary	(3 items)
UNNotificationExtensionCategory		String	NotificationContentCategory
UNNotificationExtensionDefaultContentHidden		Boolean	YES
UNNotificationExtensionInitialContentSizeRatio		Number	0.2
NSExtensionMainStoryboard		String	MainInterface
NSExtensionPointIdentifier		String	com.apple.usernotifications.content-extension

# Content Extension



# Content Extension

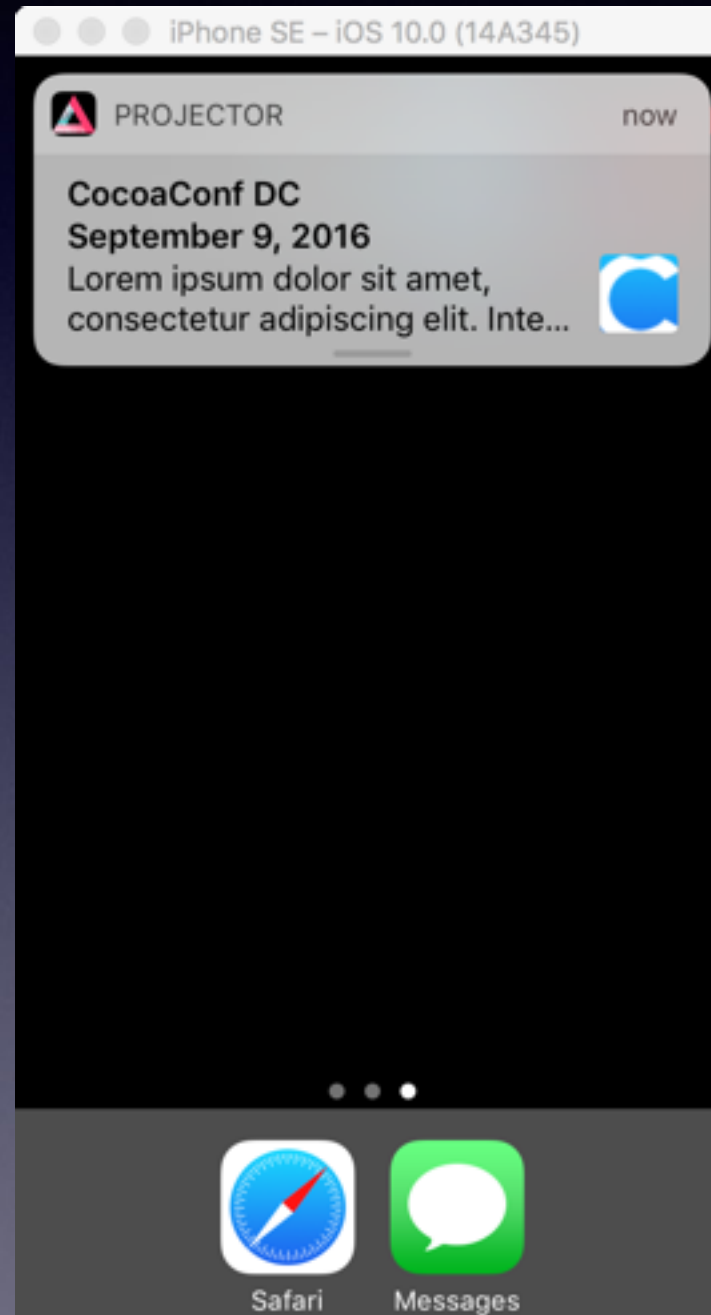


# Content Extension

```
func didReceive(_ notification: UNNotification) {  
    self.label?.text = notification.request.content.body  
  
    let content = notification.request.content  
    guard let attachment = content.attachments.first else { return }  
    if attachment.url.startAccessingSecurityScopedResource() {  
        imageView.image = UIImage(contentsOfFile: attachment.url.path)  
        attachment.url.stopAccessingSecurityScopedResource()  
    }  
}
```

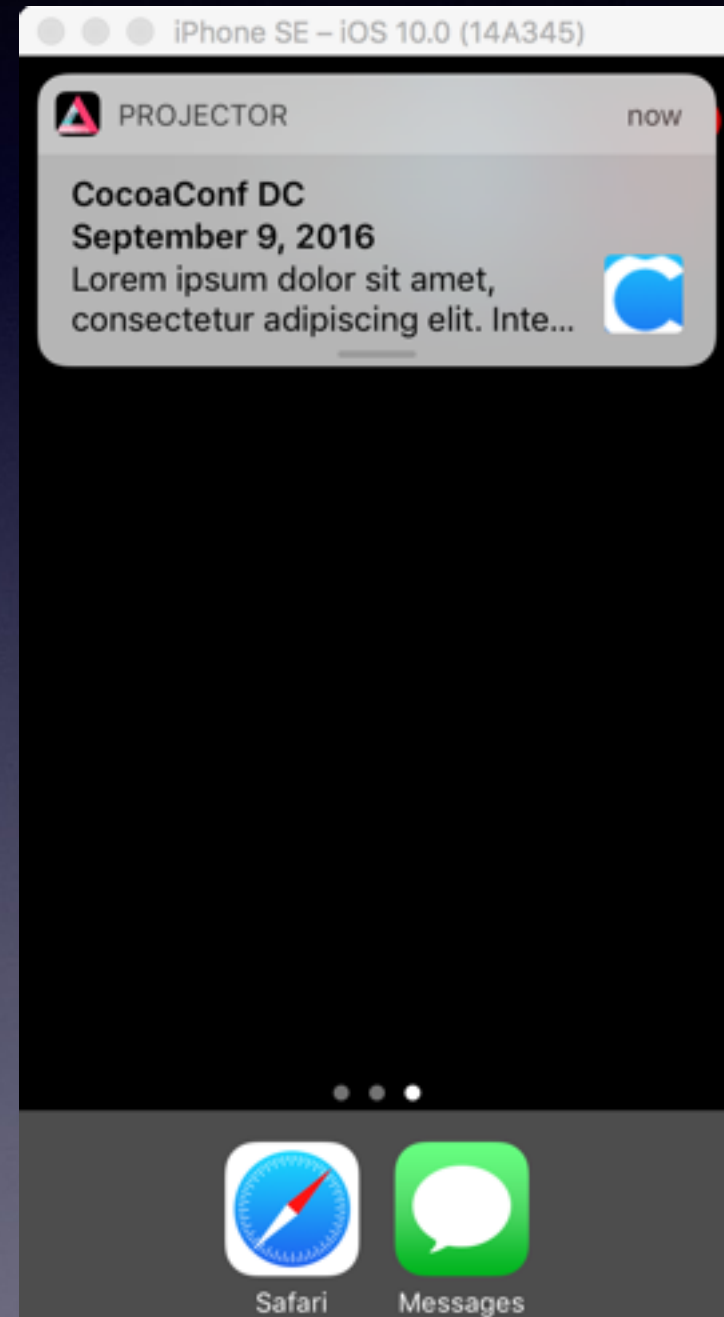


# Content Extension



# Content Extension

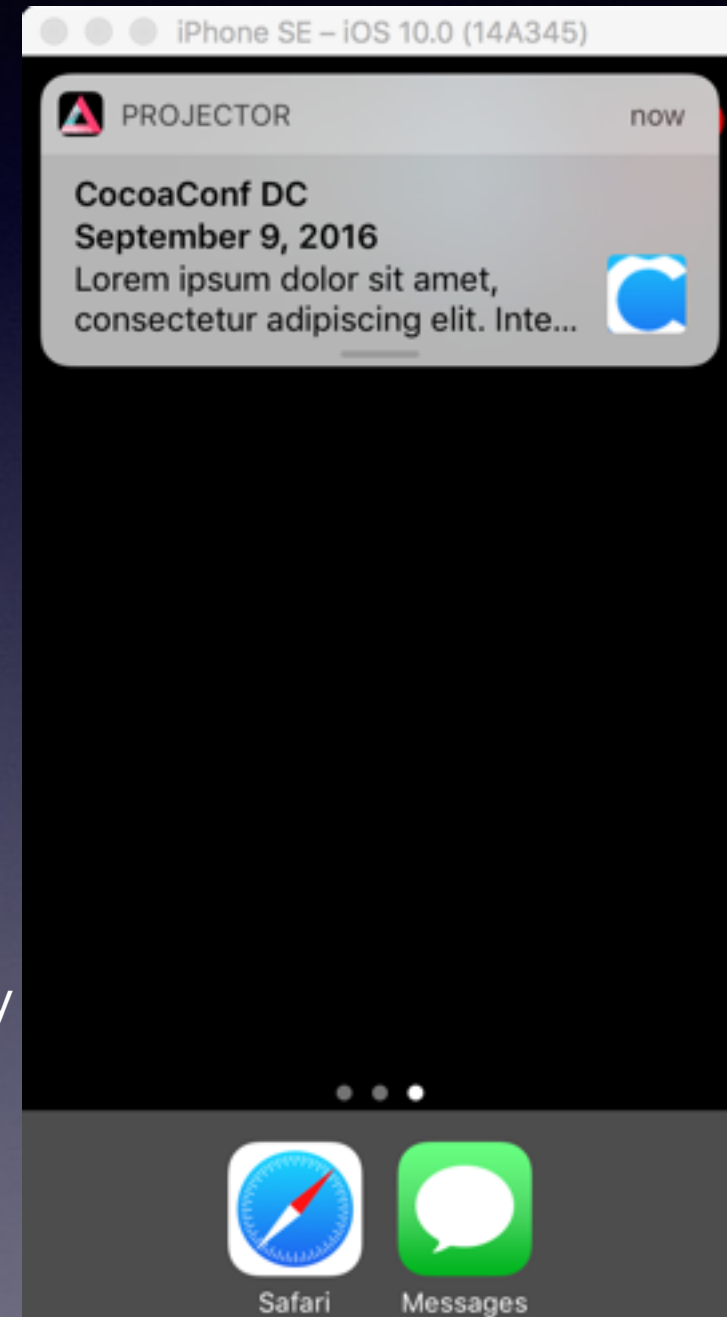
```
class UNNotificationAttachment {  
    var identifier: String { get }  
    var url: URL { get }  
    var type: String { get }  
    init(identifier: String,  
        url URL: URL,  
        options: [AnyHashable : Any]?)  
}
```



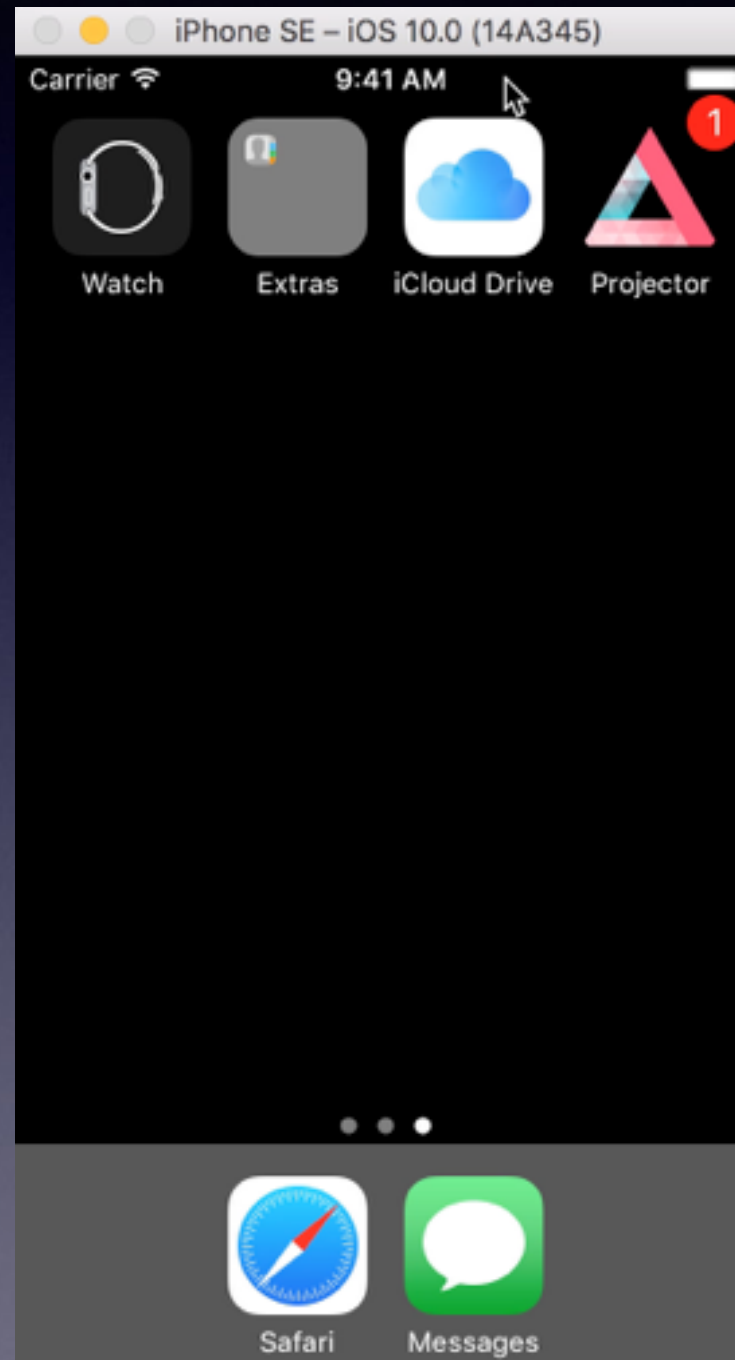
# Content Extension

```
class UNNotificationAttachment {  
    var identifier: String { get }  
    var url: URL { get }  
    var type: String { get }  
    init(identifier: String,  
        url URL: URL,  
        options: [AnyHashable : Any]?)  
}
```

```
UNNotificationAttachmentOptionsTypeHintKey  
UNNotificationAttachmentOptionsThumbnailHiddenKey  
UNNotificationAttachmentOptionsThumbnailClippingRectKey  
UNNotificationAttachmentOptionsThumbnailTimeKey
```

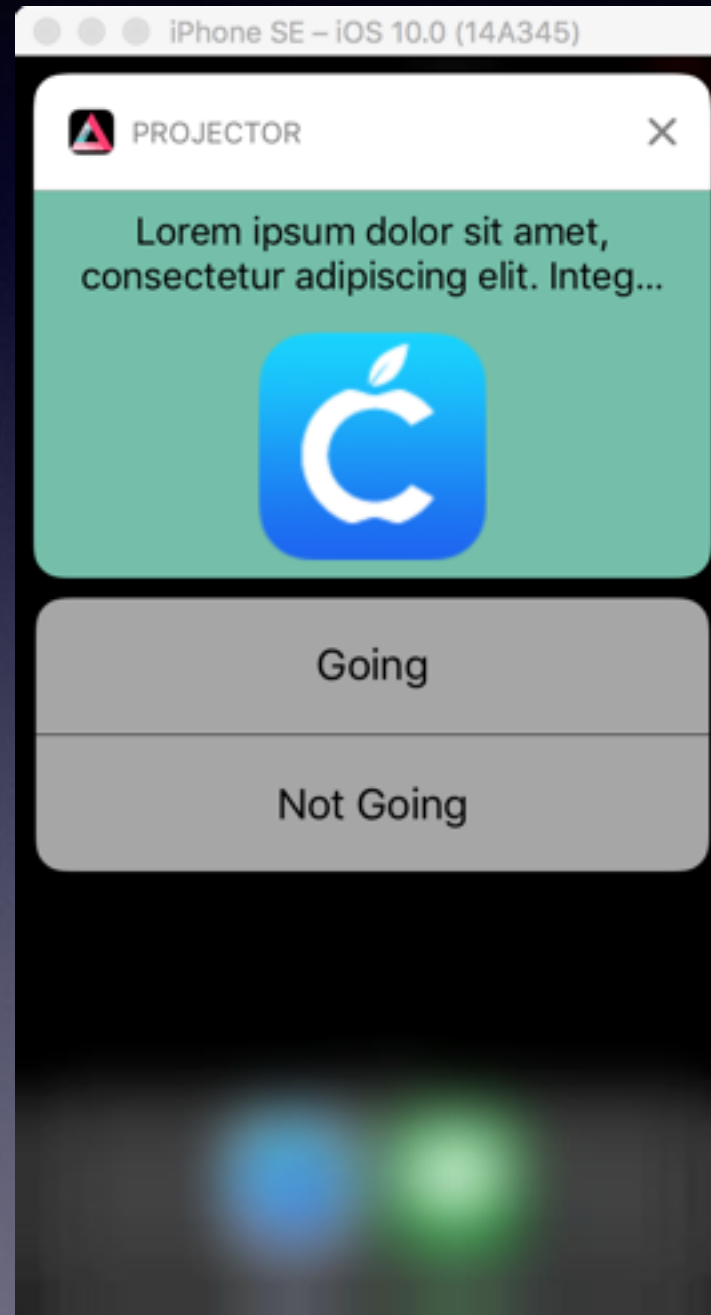


# Content Extension

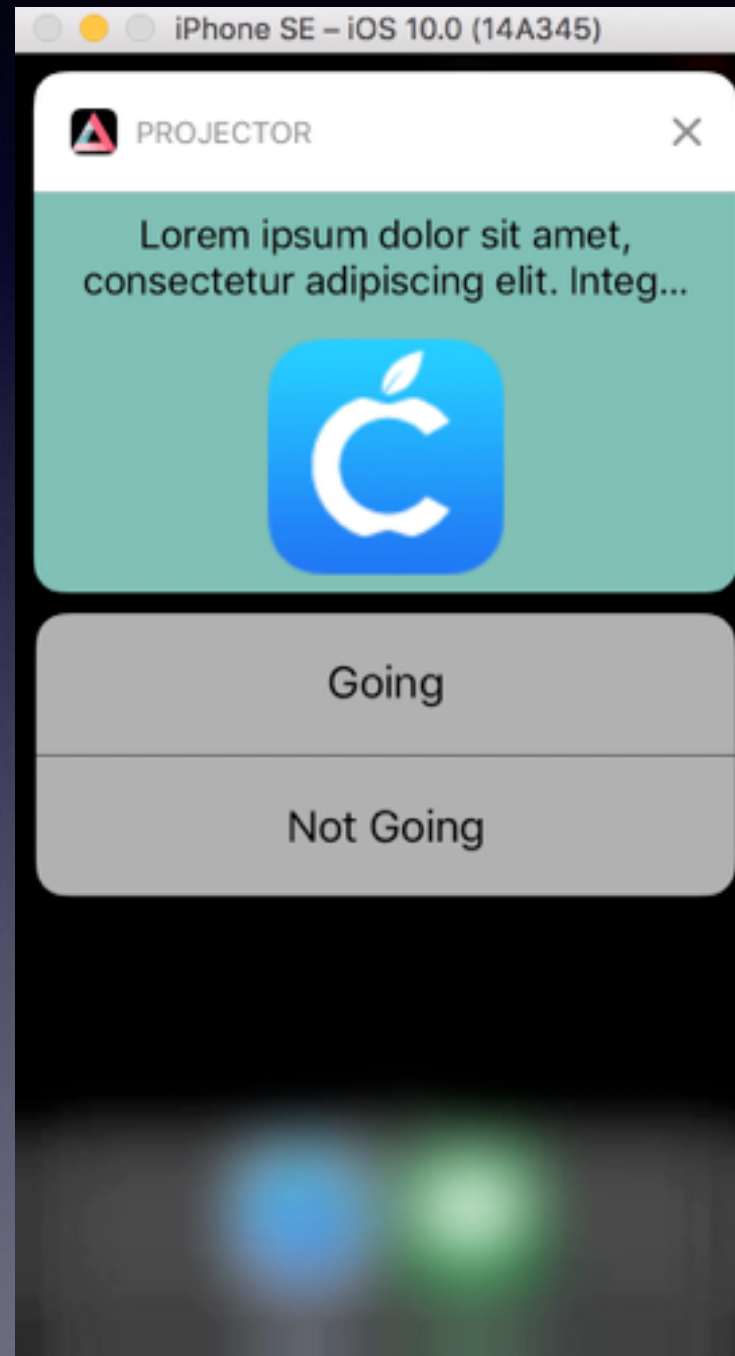




# Content Extension



# Content Extension



# Content Extension

```
func didReceive(_ response: UNNotificationResponse,
               completionHandler completion:
               @escaping (UNNotificationContentExtensionResponseOption) -> Void) {

    switch response.actionIdentifier {
    case "going":
        self.label?.text = "See You There"
        Thread.sleep(until: Date().addingTimeInterval(5.0))
    case "notGoing":
        self.label?.text = "Sorry To Hear That"
        Thread.sleep(until: Date().addingTimeInterval(2.0))
    default:
        break
    }
    completion(.dismiss)
}
```

# Content Extension

```
func didReceive(_ response: UNNotificationResponse,
               completionHandler completion:
               @escaping (UNNotificationContentExtensionResponseOption) -> Void) {

    switch response.actionIdentifier {
    case "going":
        self.label?.text = "See You There"
Thread.sleep(until: Date().addingTimeInterval(5.0))
    case "notGoing":
        self.label?.text = "Sorry To Hear That"
Thread.sleep(until: Date().addingTimeInterval(2.0))
    default:
        break
    }
    completion(.dismiss)
}
```



# Content Extension

```
func didReceive(_ response: UNNotificationResponse,
               completionHandler completion:
               @escaping (UNNotificationContentExtensionResponseOption) -> Void) {

    switch response.actionIdentifier {
    case "going":
        self.label?.text = "See You There"
Thread.sleep(until: Date().addingTimeInterval(5.0))
    case "notGoing":
        self.label?.text = "Sorry To Hear That"
Thread.sleep(until: Date().addingTimeInterval(2.0))
    default:
        break
    }
    completion(.dismiss)
}

enum UNNotificationContentExtensionResponseOption {
    case doNotDismiss
    case dismiss
    case dismissAndForwardAction
}
```

# Links

- WWDC 2016
  - 707 Introduction to Notifications
    - <https://developer.apple.com/videos/play/wwdc2016/707/>
  - 708 Advanced Notifications
    - <https://developer.apple.com/videos/play/wwdc2016/708/>
  - 724 What's New in the APNS
    - <https://developer.apple.com/videos/play/wwdc2016/724/>