What’s New in User Notifications

Session 710

Kritarth Jain, iOS Notifications
Teja Kondapalli, iOS Notifications
Grouped notifications
Notification content extensions
Notification management
Provisional authorization
Critical alerts
Grouped Notifications
Grouped Notifications
Grouped Notifications

Automatic grouping
Grouped Notifications

Automatic grouping

Thread identifier
let content = UNMutableNotificationContent()
content.title = "New Photo"
content.body = "Jane Doe posted a new photo"
content.threadIdentifier = "thread-identifier"
// Grouped Notifications
// Thread identifier

let content = UNMutableNotificationContent()
content.title = "New Photo"
content.body = "Jane Doe posted a new photo"
content.threadIdentifier = "thread-identifier"
// Grouped Notifications
// Thread identifier

{
  "aps": {
    "alert": {
      "title": "New Photo",
      "body": "Jane Doe posted a new photo",
      "thread-id": "thread-identifier"
    }
  }
}
// Grouped Notifications
// Thread identifier

{
    "aps" : {
        "alert" : {
            "title" : "New Photo",
            "body" : "Jane Doe posted a new photo",
            "thread-id" : "thread-identifier",
        },
    },
}
Grouped Notifications
Thread identifier
Grouped Notifications
Thread identifier

Content extension forwarding
Grouped Notifications
Thread identifier

Content extension forwarding

Hidden preview grouping
Grouped Notifications
Grouped Notifications

- Notification
  - No thread identifier

- App Group
Grouped Notifications

- Notification
  - No thread identifier
  - Thread identifier

  → App Group
  → Custom Group
Grouped Notifications

- Notification
  - No thread identifier → App Group
  - Thread identifier → Custom Group
    - Custom Group
    - Custom Group
    - Custom Group

Same app
Sarah Jones
How's the presentation coming along?
1 more message

John Appleseed
Hey
New Photo
Nancy Steele just posted a new photo
4 more notifications
New Photo
Nancy Steele just posted a new photo
4 more notifications
New Photo
Nancy Steele just posted a new photo
4 more notifications
Grouped Notifications

Using Grouped Notifications | Hall 3 | Wednesday 3:00PM
Grouped Notifications

Using Grouped Notifications

Hall 3

Wednesday 3:00PM
Grouped Notifications

Summary

Automatic grouping
# Grouped Notifications

## Summary

Automatic grouping

Thread identifier

<table>
<thead>
<tr>
<th>Using Grouped Notifications</th>
<th>Hall 3</th>
<th>Wednesday 3:00PM</th>
</tr>
</thead>
</table>
Grouped Notifications

Summary

Automatic grouping

Thread identifier

User setting

Using Grouped Notifications

Hall 3

Wednesday 3:00PM
Grouped Notifications

Summary

Automatic grouping

Thread identifier

User setting

Summary text

Using Grouped Notifications Hall 3 Wednesday 3:00PM
Notification Content Extensions
New Photo
Nancy Steele just posted a new photo
Notification Content Extensions

Choose a template for your new target:

iOS  macOS  Cross-platform

Application Extension

Content Blocker Extension  Custom Keyboard Extension  File Provider Extension  File Provider UI Extension  iMessage Extension

Intents Extension  Intents UI Extension  Message Filter Extension  Network Extension

Notification Service Extension  Password Manager  Photo Editing Extension  Quick Look Preview Extension  Share Extension

Cancel

Previous  Next
// Notification Content Extensions

import UserNotifications
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ notification: UNNotification) {
        // Set up content extension view
    }
}

}
import UserNotifications
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ notification: UNNotification) {
        // Set up content extension view
    }
}
# Notification Content Extensions

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Property List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Bundle display name</td>
<td>String</td>
<td>NotificationExtension</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>XPC!</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>?? ??</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>NSExtension</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSExtensionMainStoryboard</td>
<td>String</td>
<td>MainInterface</td>
</tr>
<tr>
<td>NSExtensionPointIdentifier</td>
<td>String</td>
<td>com.apple.usernotifications.content-extension</td>
</tr>
<tr>
<td>NSExtensionAttributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNNotificationExtensionCategory</td>
<td>String</td>
<td>extension-example</td>
</tr>
<tr>
<td>UNNotificationExtensionInitialContentSizeRatio</td>
<td>Number</td>
<td>0.8</td>
</tr>
<tr>
<td>UNNotificationExtensionDefaultContentHidden</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionOverridesDefaultTitle</td>
<td>Boolean</td>
<td>YES</td>
</tr>
</tbody>
</table>
## Notification Content Extensions

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Property List</td>
<td>Dictionary</td>
<td>(11 items)</td>
</tr>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Bundle display name</td>
<td>String</td>
<td>NotificationExtension</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>XPC!</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>????</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>NSExtension</td>
<td>Dictionary</td>
<td>(3 items)</td>
</tr>
<tr>
<td>NSExtensionMainStoryboard</td>
<td>String</td>
<td>MainInterface</td>
</tr>
<tr>
<td>NSExtensionPointIdentifier</td>
<td>String</td>
<td>com.apple.usernotifications.content-extension</td>
</tr>
<tr>
<td>NSExtensionAttributes</td>
<td>Dictionary</td>
<td>(4 items)</td>
</tr>
<tr>
<td>UNNotificationExtensionCategory</td>
<td>String</td>
<td>extension-example</td>
</tr>
<tr>
<td>UNNotificationExtensionInitialContentSizeRatio</td>
<td>Number</td>
<td>0.8</td>
</tr>
<tr>
<td>UNNotificationExtensionDefaultContentHidden</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionOverridesDefaultTitle</td>
<td>Boolean</td>
<td>YES</td>
</tr>
</tbody>
</table>
## Notification Content Extensions

### Information Property List

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Bundle display name</td>
<td>String</td>
<td>NotificationExtension</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>XPC!</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>?? ??</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
</tbody>
</table>

### NSExtension

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSExtensionMainStoryboard</td>
<td>String</td>
<td>MainInterface</td>
</tr>
<tr>
<td>NSExtensionPointIdentifier</td>
<td>String</td>
<td>com.apple.usernotifications.content-extension</td>
</tr>
</tbody>
</table>

### NSExtensionAttributes

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNNotificationExtensionCategory</td>
<td>String</td>
<td>extension-example</td>
</tr>
<tr>
<td>UNNotificationExtensionInitialContentSizeHidden</td>
<td>Number</td>
<td>0.8</td>
</tr>
<tr>
<td>UNNotificationExtensionDefaultContentHidden</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionOverridesDefaultTitle</td>
<td>Boolean</td>
<td>YES</td>
</tr>
</tbody>
</table>
Notification Content Extensions

Actions
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

Like

Comment
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

Like

Comment
// Notification Content Extensions

// Actions

let likeAction = UNNotificationAction(identifier: "like-action",
    title: "Like",
    options: [])

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
    title: "Comment",
    options: [],
    textInputButtonTitle: "Comment",
    textInputPlaceholder: "Type here...")

let category = UNNotificationCategory(identifier: "extension-example",
    actions: [ likeAction, commentAction ],
    intentIdentifiers: [],
    options: [])

// Notification Content Extensions

// Actions

let likeAction = UNNotificationAction(identifier: "like-action",
                                      title: "Like",
                                      options: [])

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
                                                    title: "Comment",
                                                    options: [],
                                                    textInputButtonTitle: "Comment",
                                                    textInputPlaceholder: "Type here..."
                                                    )

let category = UNNotificationCategory(identifier: "extension-example",
                                        actions: [likeAction, commentAction],
                                        intentIdentifiers: [],
                                        options: [])
// Notification Content Extensions
// Actions

let likeAction = UNNotificationAction(identifier: "like-action",
  title: "Like",
  options: [])

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
  title: "Comment",
  options: [],
  textInputButtonTitle: "Comment",
  textInputPlaceholder: "Type here...")

let category = UNNotificationCategory(identifier: "extension-example",
  actions: [ likeAction, commentAction ],
  intentIdentifiers: [],
  options: [])

// Notification Content Extensions
// Actions

let likeAction = UNNotificationAction(identifier: "like-action",
    title: "Like",
    options: [])

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
    title: "Comment",
    options: [],
    textInputButtonTitle: "Comment",
    textInputPlaceholder: "Type here...")

let category = UNNotificationCategory(identifier: "extension-example",
    actions: [ likeAction, commentAction ],
    intentIdentifiers: [],
    options: [])
// Notification Content Extensions

// Actions

let likeAction = UNNotificationAction(identifier: "like-action",
    title: "Like",
    options: [])

let commentAction = UNTextInputNotificationAction(identifier: "comment-action",
    title: "Comment",
    options: [],
    textInputButtonTitle: "Comment",
    textInputPlaceholder: "Type here..."
)

let category = UNNotificationCategory(identifier: "extension-example",
    actions: [likeAction, commentAction],
    intentIdentifiers: [],
    options: [])
// Notification Content Extensions
// Actions

let category = UNNotificationCategory(identifier: "extension-example",
    actions: [ likeAction, commentAction ],
    intentIdentifiers: [],
    options: [])

UNUserNotificationCenter.current().setNotificationCategories([ category ])
// Notification Content Extensions

// Actions

let category = UNNotificationCategory(identifier: "extension-example",
    actions: [ likeAction, commentAction ],
    intentIdentifiers: [],
    options: [])

UNUserNotificationCenter.current().setNotificationCategories([[ category ]])
New Photo
Nancy Steele just posted a new photo
import UserNotifications

extension AppDelegate: UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter,
        didReceive response: UNNotificationResponse,
        withCompletionHandler completionHandler: () -> Void) {
        // Handle action response here
    }

}
// Notification Content Extensions
// Actions handling

import UserNotifications

extension AppDelegate: UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter,
                               didReceive response: UNNotificationResponse,
                               withCompletionHandler completionHandler: () -> Void) {
        // Handle action response here
    }
}

// Notification Content Extensions
// Actions handling

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            likeLabel?.text = "You liked this photo"
            likedPhoto()
        }
        completion(.doNotDismiss)
    }
}
// Notification Content Extensions
// Actions handling

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            likeLabel?.text = "You liked this photo"
            likedPhoto()
        }
        completion(.doNotDismiss)
    }
}
// Notification Content Extensions
// Actions handling

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            likeLabel?.text = "You liked this photo"
            likedPhoto()
        }

        completion(.doNotDismiss)
    }
}
// Notification Content Extensions
// Actions handling

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            likeLabel?.text = "You liked this photo"
            likedPhoto()
        }
        completion(.doNotDismiss)
    }
}
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            likeLabel?.text = "You liked this photo"
            likedPhoto()
        }
        completion(.dismiss)
    }
}
// Notification Content Extensions
// Actions handling

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeLabel: UILabel?

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            likeLabel?.text = "You liked this photo"
            likedPhoto()
        }
        completion(.dismissAndForward)
    }
}
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
Nancy Steele, Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

You liked this photo

Like

Comment
Enjoying the bus ride in Lisbon 🌟
You liked this photo
Like
Comment
Notification Content Extensions

Actions
Notification Content Extensions

Actions

Not dynamic
Notification Content Extensions

Actions

Not dynamic

Tied to category
// Notification Content Extensions
// Actions API

extension NSExtensionContext {

    @available(iOS 12.0, *)
    var notificationActions: [UNNotificationAction]

}
// Notification Content Extensions
// Actions API

extension NSExtensionContext {

   @available(iOS 12.0, *)
   var notificationActions: [UNNotificationAction]

}
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

You liked this photo

Like

Comment
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
You liked this photo

Like

Comment
Nancy Steele, Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
You liked this photo

Unlike
Comment
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            // Update state...
            let unlikeAction = UNNotificationAction(identifier: "unlike-action",
                                                   title: "Unlike", options: [])
            let currentActions = extensionContext?.notificationActions
            let commentAction = currentActions![1]
            let newActions = [ unlikeAction, commentAction ]
            extensionContext?.notificationActions = newActions
        }
    }
}
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            // Update state...
            let unlikeAction = UNNotificationAction(identifier: "unlike-action",
                title: "Unlike", options: [])
            let currentActions = extensionContext?.notificationActions
            let commentAction = currentActions![1]
            let newActions = [ unlikeAction, commentAction ]
            extensionContext?.notificationActions = newActions
        }
    }
}
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            // Update state...

            let unlikeAction = UNNotificationAction(identifier: "unlike-action",
                                                     title: "Unlike", options: [])

            let currentActions = extensionContext?.notificationActions
            let commentAction = currentActions![1]
            let newActions = [ unlikeAction, commentAction ]
            extensionContext?.notificationActions = newActions
        }
    }
}
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            // Update state...
            let unlikeAction = UNNotificationAction(identifier: "unlike-action",
                title: "Unlike", options: [])

            let currentActions = extensionContext?.notificationActions
            let commentAction = currentActions![1]

            let newActions = [ unlikeAction, commentAction ]
            extensionContext?.notificationActions = newActions
        }
    }
}
class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            // Update state...
            let unlikeAction = UNNotificationAction(identifier: "unlike-action",
                                                    title: "Unlike", options: [])

            let currentActions = extensionContext?.notificationActions
            let commentAction = currentActions![1]

            let newActions = [ unlikeAction, commentAction ]
            extensionContext?.notificationActions = newActions
        }
    }
}
// Notification Content Extensions

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        if response.actionIdentifier == "like-action" {
            // Update state...
            let unlikeAction = UNNotificationAction(identifier: "unlike-action",
                                                    title: "Unlike", options: [])
            let currentActions = extensionContext?.notificationActions
            let commentAction = currentActions![1]
            let newActions = [ unlikeAction, commentAction ]
            extensionContext?.notificationActions = newActions
        }
    }
}

Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon ☀️

Like
Comment
Nancy Steele, Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
func didReceive(_ notification: UNNotification) {
    // Set up content extension view...
    extensionContext?.notificationActions = newActions
}
func didReceive(_ notification: UNNotification) {
    // Set up content extension view...
    extensionContext?.notificationActions = newActions
}
func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
    if response.actionIdentifier == "rate-action" {
        extensionContext?.notificationActions = [ oneStarAction, twoStarAction, threeStarAction ]
    }
}
func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
    if response.actionIdentifier == "rate-action" {
        extensionContext?.notificationActions = [oneStarAction, twoStarAction, threeStarAction]
    }
}
Notification Content Extensions

Actions

Remove all notification actions

```swift
func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
    if response.actionIdentifier == "done-action" {
        extensionContext?.notificationActions = []
    }
}
```
Notification Content Extensions

Actions

Remove all notification actions

```swift
func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
    if response.actionIdentifier == "done-action" {
        extensionContext?.notificationActions = []
    }
}
```
extension NSExtensionContext {

    @available(iOS 12.0, *)
    var notificationActions: [UNNotificationAction]

}
// Notification Content Extensions
// Actions API

extension NSExtensionContext {

    @available(iOS 12.0, *)
    var notificationActions: [UNNotificationAction]

}
Notification Content Extensions

User interactions
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon ☀
You liked this photo

Like
Comment
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
You liked this photo

Like
Comment
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
You liked this photo

Like

Comment
Notification Content Extensions
User interaction

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Property List</td>
<td>Dictionary</td>
<td>(11 items)</td>
</tr>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Bundle display name</td>
<td>String</td>
<td>NotificationExtension</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>XPCI</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>???.?</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>NSExtension</td>
<td>Dictionary</td>
<td>(3 items)</td>
</tr>
<tr>
<td>NSExtensionMainStoryboard</td>
<td>String</td>
<td>MainInterface</td>
</tr>
<tr>
<td>NSExtensionPointIdentifier</td>
<td>String</td>
<td>com.apple.usernotifications.content-extension</td>
</tr>
<tr>
<td>NSExtensionAttributes</td>
<td>Dictionary</td>
<td>(5 items)</td>
</tr>
<tr>
<td>UNNotificationExtensionCategory</td>
<td>String</td>
<td>extension-example</td>
</tr>
<tr>
<td>UNNotificationExtensionInitialContentSizeRatio</td>
<td>Number</td>
<td>0.8</td>
</tr>
<tr>
<td>UNNotificationExtensionDefaultContentHidden</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionOverridesDefaultTitle</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionUserInteractionEnabled</td>
<td>Boolean</td>
<td>YES</td>
</tr>
</tbody>
</table>
## Notification Content Extensions

### User interaction

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼ Information Property List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Localization native development region</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Bundle display name</td>
<td>String</td>
<td>NotificationExtension</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>XPC!</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>?????</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>▼ NSExtension</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSExtensionMainStoryboard</td>
<td>String</td>
<td>MainInterface</td>
</tr>
<tr>
<td>NSExtensionPointIdentifier</td>
<td>String</td>
<td>com.apple.usernotifications.content-extension</td>
</tr>
<tr>
<td>▼ NSExtensionAttributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNNotificationExtensionCategory</td>
<td>String</td>
<td>extension-example</td>
</tr>
<tr>
<td>UNNotificationExtensionInitialContentSizeRatio</td>
<td>Number</td>
<td>0.8</td>
</tr>
<tr>
<td>UNNotificationExtensionDefaultContentSizeHidden</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionOverridesDefaultTitle</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>UNNotificationExtensionUserInteractionEnabled</td>
<td>Boolean</td>
<td>YES</td>
</tr>
</tbody>
</table>
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌞
Nancy Steele, Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var likeButton: UIButton?
    ...

    likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
    ...

    @objc func likeButtonTapped() {
        likeButton?.setTitle("♥", for: .normal)
        likedPhoto() 
    }
}
// Notification Content Extensions
// User interaction

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

@IBOutlet var likeButton: UIButton?

...

likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
...

@objc func likeButtonTapped() {
    likeButton?.setTitle("♥", for: .normal)
    likedPhoto()
}

}
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var likeButton: UIButton?
    ...
    likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
    ...
    @objc func likeButtonTapped() {
        likeButton?.setTitle("♥", for: .normal)
        likedPhoto()
    }
}
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var likeButton: UIButton?
    ...

    likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
    ...

    @objc func likeButtonTapped() {
        likeButton?.setTitle("♥", for: .normal)
        likedPhoto()
    }
}
Notification Content Extensions
User interaction

UNNotificationExtensionUserInteractionEnabled
Notification Content Extensions

Launch application
New Photo

Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

Comment
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌞
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌞

Comment

View in App
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon ☀️

See all comments in app

Comment
// Notification Content Extensions

extension NSExtensionContext {

    @available(iOS 12.0, *)
    func performNotificationDefaultAction()

}
extension NSExtensionContext {

    @available(iOS 12.0, *)
    func performNotificationDefaultAction()

}
Notification Content Extensions
Default action
Notification Content Extensions

Default action

Launches the application
Notification Content Extensions

Default action

Launches the application

UNNotificationResponse

```swift
func userNotificationCenter(_ center: UNUserNotificationCenter, 
didReceive response: UNNotificationResponse, 
withCompletionHandler completionHandler: () -> Void) {
    // Handle action response
}
```
Notification Content Extensions

Default action

Launches the application

```swift
func userNotificationCenter(_ center: UNUserNotificationCenter,
    didReceive response: UNNotificationResponse,
    withCompletionHandler completionHandler: () -> Void) {
    // Handle action response
}
```

UNNotificationResponse
UNNotificationDefaultActionIdentifier
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌞

See all comments in app

Comment
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var allCommentsButton: UIButton?
    ...
    allCommentsButton?.addTarget(self, action: #selector(launchApp), for: .touchUpInside)
    ...
    @objc func launchApp() {
        extensionContext?.performNotificationDefaultAction()
    }
}

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {
    @IBOutlet var allCommentsButton: UIButton?
    ...
    allCommentsButton?.addTarget(self, action: #selector(launchApp), for: .touchUpInside)
    ...
    @objc func launchApp() {
        extensionContext?.performNotificationDefaultAction()
    }

}
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var allCommentsButton: UIButton?

    allCommentsButton?.addTarget(self, action: #selector(launchApp), for: .touchUpInside)

    @objc func launchApp() {
        extensionContext?.performNotificationDefaultAction()
    }

}
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var allCommentsButton: UIButton?

    allCommentsButton?.addTarget(self, action: #selector(launchApp), for: .touchUpInside)

    @objc func launchApp() {
        extensionContext?.performNotificationDefaultAction()
    }
}
// Notification Content Extensions
// Default action

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var allCommentsButton: UIButton?
    ...

    allCommentsButton?.addTarget(self, action: #selector(launchApp), for: .touchUpInside)
    ...

    @objc func launchApp() {
        extensionContext?.performNotificationDefaultAction()
    }
}

}
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟
See all comments in app

Comment
Notification Content Extensions

Dismiss content extension view
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

Comment

Done
// Notification Content Extensions

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        // Handle action...
        completion(.dismiss)
    }
}

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    func didReceive(_ response: UNNotificationResponse, completionHandler completion: (UNNotificationContentExtensionResponseOption) -> Void) {
        // Handle action...
        completion(.dismiss)
    }
}
Nancy Steele  Lisbon, Portugal

Enjoying the bus ride in Lisbon 🌟

Comment
// Notification Content Extensions

extension NSExtensionContext {

    @available(iOS 12.0, *)
    func dismissNotificationContentExtension()

}
extension NSExtensionContext {

    // Available from iOS 12.0
    @available(iOS 12.0, *)
    func dismissNotificationContentExtension()

}
// Notification Content Extensions
// User interaction

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var likeButton: UIButton?
    ...
    likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
    ...
    @objc func likeButtonTapped() {
        likedPhoto()
        extensionContext?.dismissNotificationContentExtension()
    }
}
import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var likeButton: UIButton?
    ...
    likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
    ...

    @objc func likeButtonTapped() {
        likedPhoto()
        extensionContext?.dismissNotificationContentExtension()
    }
}

// Notification Content Extensions
// User interaction

import UserNotificationsUI

class NotificationViewController: UIViewController, UNNotificationContentExtension {

    @IBOutlet var likeButton: UIButton?
    ...

    likeButton?.addTarget(self, action: #selector(likeButtonTapped), for: .touchUpInside)
    ...

    @objc func likeButtonTapped() {
        likedPhoto()
        extensionContext?.dismissNotificationContentExtension()
    }

}
Enjoying the bus ride in Lisbon 🌟
Notification Content Extensions

Dismiss content extension view
Notification Content Extensions
Dismiss content extension view

Does not withdraw the notification
Notification Content Extensions
Dismiss content extension view

Does not withdraw the notification

class UNUserNotificationCenter {

    func removeDeliveredNotifications(withIdentifiers identifiers: [String])

}
Notification Content Extensions

Dismiss content extension view

Does not withdraw the notification

class UNUserNotificationCenter {

    func removeDeliveredNotifications(withIdentifiers identifiers: [String])

}
Notification Content Extensions

Summary
Notification Content Extensions

Summary
Notification Content Extensions

Summary

Notification actions

extension NSExtensionContext {
    var notificationActions: [UNNotificationAction]
}

```swift
extension NSExtensionContext {
    var notificationActions: [UNNotificationAction]
}
```
Notification Content Extensions

Summary

Notification actions

User interaction enabled in extension view
Notification Content Extensions

Summary

Notification actions

User interaction enabled in extension view

Perform default action to launch app

default_action()
Notification Content Extensions

Summary

Notification actions

User interaction enabled in extension view

Perform default action to launch app

Dismiss content extension view

```swift
extension NSExtensionContext {
    func dismissNotificationContentExtension()
}
```
Notification Content Extensions

Summary

Notification actions

User interaction enabled in extension view

Perform default action to launch app

Dismiss content extension view
Notification Management
The New York Times
Are pizza boxes recyclable? How about coffee cups? Here are six things you’re recycling wrong.

Chase Sapphire Reserve
Apple Park Caffe
$700

b_iWhiz sent you a photo.

You completed Bihter's request for $70.00 - May 30

Food Network
Bad news: Your water bottle is actually really dirty, and cleaning it every few days isn’t enough. 😞

Christopher Kimball's Milk Street Radio
Nigella Lawson Speaks Out on Hash, Traybakes, Salad Snobs and Kierkegaard (corrected) is available.

NEWS
8:56 AM

PODCASTS
8:29 AM
The New York Times
Are pizza boxes recyclable? How about coffee cups? Here are six things you’re recycling wrong.

Chase Sapphire Reserve
Apple Park Caffe
$700

b_jWhiz sent you a photo.

You completed Bihter's request for $70.00 - May 30

Food Network
Bad news: Your water bottle is actually really dirty, and cleaning it every few days isn’t enough.

Christopher Kimball’s Milk Street Radio
Nigella Lawson Speaks Out on Hash, Traybakes, Salad Snobs and Kierkegaard (corrected) is available.
NEW

Wednesday, June 6

Podcasts

Christopher Kimball's Milk Street Radio
5 new episodes are available.
Podcasts
Manage notifications

Deliver Quietly
Turn Off...

Settings
Cancel

Quiet notifications will appear in notification center and badge the app icon, but will not appear on the lock screen, present banners, or play a sound.
Notification Management
Deliver Prominently
Notification Management

Deliver Quietly
Podcasts
Manage notifications

Deliver Quietly

Turn Off...

Quiet notifications will appear in notification center and badge the app icon, but will not appear on the lock screen, present banners, or play a sound.

Settings

Cancel
Podcasts
Manage notifications

Deliver Quietly

Are you sure you want to turn off all notifications from Podcasts?

Turn off all notifications
Configure in Podcasts...

Cancel
import UIKit
import UserNotifications

class AppDelegate: UIApplicationDelegate, UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter,
                            openSettingsFor notification: UNNotification?) {

    }

}
import UIKit
import UserNotifications

class AppDelegate: UIApplicationDelegate, UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter, openSettingsFor notification: UNNotification? ) {
    }
}

import UIKit
import UserNotifications

class AppDelegate: UIApplicationDelegate, UNUserNotificationCenterDelegate {

    func userNotificationCenter(_ center: UNUserNotificationCenter,
                                openSettingsFor notification: UNNotification?) {

    }

}
Notification Management

Summary
Notification Management

Summary

Make notification content relevant
Notification Management

Summary

Make notification content relevant

Group notifications
Notification Management

Summary

Make notification content relevant

Group notifications

Provide custom settings view
Provisional Authorization
“PersonalTracker” Would Like to Send You Notifications

Notifications may include alerts, sounds, and icon badges. These can be configured in Settings.

Don’t Allow  Allow
Provisional Authorization
Provisional Authorization

Automatic trial
Provisional Authorization

Automatic trial

No authorization prompt
Provisional Authorization

Automatic trial

No authorization prompt

Notifications delivered quietly
// Requesting Provisional Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options:[.badge, .sound, .alert, .provisional]) {
}
let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options: [.badge, .sound, .alert, .provisional]) {
}
// Requesting Provisional Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options: [.badge, .sound, .alert, .provisional]) { }
// Requesting Provisional Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options: [.badge, .sound, .alert, .provisional]) {

}
// Requesting Provisional Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options: [.badge, .sound, .alert, .provisional]) {
}
Provisional Authorization

Summary
Provisional Authorization

Summary

Make notification content relevant
Provisional Authorization

Summary

Make notification content relevant

Use provision option as a qualifier
Critical Alerts
WARNING: LOW BLOOD SUGAR
Glucose level at 57.
Critical Alerts

Medical and health
Critical Alerts

Medical and health

Home and security
Critical Alerts

Medical and health
Home and security
Public safety
Critical Alerts

Medical and health
Home and security
Public safety

**Key:** Alerts that require the user to take action
Critical Alerts
Critical Alerts

Bypass Do Not Disturb and ringer switch
Critical Alerts

Bypass Do Not Disturb and ringer switch

Play a sound
Critical Alerts

Bypass Do Not Disturb and ringer switch

Play a sound

Disruptive
Critical Alerts
Entitlement

https://developer.apple.com/contact/request/
notifications-critical-alerts-entitlement/
WARNING: LOW BLOOD SUGAR
Glucose level at 57.
Allow Critical Alerts

Critical alerts appear on the lock screen and play a sound even if Do Not Disturb is on or iPhone is muted.

Allow Notifications

ALERTS

Lock Screen
Notification Center
Banners

Banner Style

Temporary

Sounds

Badges

OPTIONS

Show Previews
When Unlocked (Default)

Notification Grouping
Automatic
Allow Critical Alerts
Show Previews: When Unlocked (Default)

Critical alerts appear on the lock screen and play a sound even if Do Not Disturb is on or iPhone is muted.

Allow Notifications
“Glucose Monitor” Would Like to Send You Critical Alerts

Critical Alerts always play a sound and appear on the lock screen even if your iPhone is muted or Do Not Disturb is on. Manage Critical Alerts in Settings.

Don’t Allow  Allow
// Requesting Critical Alerts Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options:[.sound, .badge, .alert, .criticalAlert]) {
}
// Requesting Critical Alerts Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(options:[.sound, .badge, .alert, .criticalAlert]) {
}
// Requesting Critical Alerts Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options: [.sound, .badge, .alert, .criticalAlert]) {
}
let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options:[.sound, .badge, .alert, .criticalAlert])
}
// Requesting Critical Alerts Authorization

let notificationCenter = UNUserNotificationCenter.current()

notificationCenter.requestAuthorization(
    options:[.sound, .badge, .alert, .criticalAlert]) {

}
// Critical alert with default sound

let content = UNMutableNotificationContent()
content.title = "WARNING: LOW BLOOD SUGAR"
content.body = "Glucose level at 57."
content.categoryIdentifier = "low-glucose-alert"
content.sound = UNNotificationSound.defaultCritical
let content = UNMutableNotificationContent()
content.title = "WARNING: LOW BLOOD SUGAR"
content.body = "Glucose level at 57."
content.categoryIdentifier = "low-glucose-alert"
content.sound = UNNotificationSound.defaultCritical
let content = UNMutableNotificationContent()
content.title = "WARNING: LOW BLOOD SUGAR"
content.body = "Glucose level at 57."
content.categoryIdentifier = "low-glucose-alert"
content.sound = UNNotificationSound.criticalSoundNamed("warning-sound" withAudioVolume: 1.00)
let content = UNMutableNotificationContent()
content.title = "WARNING: LOW BLOOD SUGAR"
content.body = "Glucose level at 57."
content.categoryIdentifier = "low-glucose-alert"
content.sound = UNNotificationSound.criticalSoundNamed("warning-sound" withAudioVolume: 1.00)
// Critical alert push payload

{
    "aps": {
        "sound": {
            "critical": 1,
            "name": "warning-sound.aiff",
            "volume": 1.0
        }
    }
}
// Critical alert push payload

{
    "aps": {
        "sound": {
            "critical": 1,
            "name": "warning-sound.aiff",
            "volume": 1.0
        }
    }
}
Critical Alerts

Summary

Alerts that require the user to take action
Critical Alerts

Summary

Alerts that require the user to take action

Disruptive
Summary
Summary
Summary

Grouped notifications
Summary

Grouped notifications

Notification content extensions
Summary

Grouped notifications

Notification content extensions

Notification management
Summary

Grouped notifications
Notification content extensions
Notification management
Provisional authorization
Summary

Grouped notifications
Notification content extensions
Notification management
Provisional authorization
Critical alerts
<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using Grouped Notifications</td>
<td>Hall 3</td>
<td>Wednesday 3:00PM</td>
</tr>
<tr>
<td>Notifications Lab</td>
<td>Technology Lab 1</td>
<td>Wednesday 4:00PM</td>
</tr>
<tr>
<td>Notifications Lab</td>
<td>Technology Lab 1</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>Designing Notifications</td>
<td>Hall 2</td>
<td>Friday 10:00AM</td>
</tr>
</tbody>
</table>

More Information

https://developer.apple.com/wwdc18/710