What’s New in Notifications

Session 720

Michele Campeotto iOS Notifications
Gokul Thirumalai Apple Push Notification Service

© 2015 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.
Today
Today

iOS Notifications
Today

iOS Notifications
• Silent Notifications
Today

iOS Notifications

- Silent Notifications
- User Notifications
Today

iOS Notifications
• Silent Notifications
• User Notifications
• Actions
iOS Notifications

• Silent Notifications
• User Notifications
• Actions
• Text input — New!
iOS Notifications

- Silent Notifications
- User Notifications
- Actions
- Text input — New!

Apple Push Notification Service
Silent Notifications
Silent Notifications

Notify the app
Silent Notifications

Notify the app
Silent Notifications

Notify the app

{ content-available: 1 }
Silent Notifications

Notify the app

{ content-available: 1 }

downloading...
Silent Notifications

Notify the app
Enabled by default

{ content-available: 1 }

downloading...
Silent Notifications

Notify the app
Enabled by default
Background App Refresh

{ content-available: 1 }

downloading...
Silent Notifications

Notify the app
Enabled by default
Background App Refresh
Best effort

{ content-available: 1 }

downloading...
User Notifications
User Notifications

Notify the user
User Notifications

Notify the user
User Notifications

Notify the user
User Notifications

Notify the user
User Notifications

Notify the user
User Notifications

Notify the user

Require user permission
User Notifications

Notify the user

Require user permission
User Notifications

Notify the user

Require user permission

Can be disabled
User Notifications

Notify the user
Require user permission
Can be disabled
User Notifications
User Notifications

Remote Notifications
User Notifications

Remote Notifications
User Notifications

Remote Notifications

{ content-available: 1 }
User Notifications

Remote Notifications
Local Notifications
User Notifications

Remote Notifications
Local Notifications
User Notifications

Remote Notifications
Local Notifications
User Notifications

Remote Notifications

Local Notifications

• Time
User Notifications

Remote Notifications

Local Notifications

• Time

• Location
User Notifications Actions
User Notifications Actions

Interactive Notifications
User Notifications Actions

Interactive Notifications
Categories
User Notifications Actions

Interactive Notifications

Categories
Apple Watch
Apple Watch

WatchKit In-Depth, Part 1

WatchKit In-Depth, Part 2
User Notifications Text Input
Text Input Action

New action
Text Input Action

New action
Text Input Action

New action
Everywhere
Text Input Action

New action
Everywhere
Text Input Action

New action

Everywhere

Multiple actions
Text Input Action

New action
Everywhere
Multiple actions
Text Input Action
Registering
let replyAction = UIMutableUserNotificationAction()
replyAction.title = "Reply"
replyAction.identifier = "comment-reply"
replyAction.activationMode = .Background
replyAction.requiresAuthentication = false
replyAction.behavior = .TextInput
let replyAction = UIMutableUserNotificationAction()
replyAction.title = "Reply"
replyAction.identifier = "comment-reply"
replyAction.activationMode = .Background
replyAction.requiresAuthentication = false
replyAction.behavior = .TextInput
Text Input Action
Registering
Text Input Action

Registering

```swift
let category = UILocalUserNotificationCategory()
category.identifier = "reply"
category.actions = [replyAction]
```
let category = UIMutableUserNotificationCategory()
category.identifier = "reply"
category.actions = [replyAction]

let settings = UIUserNotificationSettings(forTypes: [.Sound, .Alert],
categories: [category])
let category = UIMutableUserNotificationCategory()
category.identifier = "reply"
category.actions = [replyAction]

let settings = UIUserNotificationSettings(forTypes: [.Sound, .Alert],
categories: [category])

UIApplication.sharedApplication().registerUserNotificationSettings(settings)
protocol UIApplicationDelegate {

    func application(application: UIApplication, 
        handleActionWithIdentifier identifier: String?, 
        forRemoteNotification notification: [NSObject : AnyObject], 
        withResponseInfo responseInfo: [NSObject : AnyObject], 
        completionHandler completionHandler: () -> Void)

}
protocol UIApplicationDelegate {

    func application(_ application: UIApplication, 
        handleActionWithIdentifier identifier: String?, 
        forRemoteNotification notification: [NSObject: AnyObject], 
        withResponseInfo responseInfo: [NSObject: AnyObject], 
        completionHandler completionHandler: () -> Void)

}
protocol UIApplicationDelegate {

    func application(application: UIApplication, 
        handleActionWithIdentifier identifier: String?, 
        forLocalNotification notification: UILocalNotification, 
        withResponseInfo responseInfo: [NSObject : AnyObject], 
        completionHandler completionHandler: () -> Void)

}
protocol UIApplicationDelegate {

    func application(application: UIApplication,
                handleActionWithIdentifier identifier: String?,
                forLocalNotification notification: UILocalNotification,
                withResponseInfo responseInfo: [NSObject : AnyObject],
                completionHandler completionHandler: () -> Void)

}
func application(application: UIApplication, 
handleActionWithIdentifier identifier: String?, 
forLocalNotification notification: UILocalNotification, 
withResponseInfo responseInfo: [ NSObject : AnyObject ], 
completionHandler completionHandler: () -> Void) {

    if identifier == "comment-reply",
    let response = responseInfo[UIUserNotificationActionResponseTypedTextKey],
    responseText = response as? String {
        viewController.appendText(responseText)
    }
}

completionHandler()
func application(application: UIApplication,
handleActionWithIdentifier identifier: String?,
forLocalNotification notification: UILocalNotification,
withResponseInfo responseInfo: [NSObject : AnyObject],
completionHandler completionHandler: () -> Void) {

if identifier == "comment-reply",
let response = responseInfo[UIUserNotificationActionResponseTypedTextKey],
responseText = response as? String {
viewController.appendText(responseText)
}

completionHandler()
}
func application(application: UIApplication, 
handleActionWithIdentifier identifier: String?, 
forLocalNotification notification: UILocalNotification, 
withResponseInfo responseInfo: [NSObject : AnyObject], completionHandler completionHandler: () -> Void) {

if identifier == "comment-reply",
    let response = responseInfo[UIUserNotificationActionResponseTypedTextKey], 
    responseText = response as? String {
        viewController.appendText(responseText)
    }
}

completionHandler()
func application(application: UIApplication, handleActionWithIdentifier identifier: String?, forLocalNotification notification: UILocalNotification, withResponseInfo responseInfo: [NSObject : AnyObject], completionHandler completionHandler: () -> Void) {

    if identifier == "comment-reply",
        let response = responseInfo[UIUserNotificationActionResponseTypedTextKey],
        responseText = response as? String {
        viewController.appendText(responseText)
    }
}

completionHandler()
Text Input Action
Handling the response

```swift
func application(application: UIApplication,
                 handleActionWithIdentifier identifier: String?,
                 forLocalNotification notification: UILocalNotification,
                 withResponseInfo responseInfo: [NSObject : AnyObject],
                 completionHandler completionHandler: () -> Void) {

    if identifier == "comment-reply",
        let response = responseInfo[UIUserNotificationActionResponseTypedTextKey],
        responseText = response as? String {
            viewController.appendText(responseText)
    }

    completionHandler()
}
```
func application(application: UIApplication,
    handleActionWithIdentifier identifier: String?,
    forLocalNotification notification: UILocalNotification,
    withResponseInfo responseInfo: [NSObject : AnyObject],
    completionHandler completionHandler: () -> Void) {

    if identifier == "comment-reply",
        let response = responseInfo[UIUserNotificationActionResponseTypedTextKey],
        responseText = response as? String {
        viewController.appendText(responseText)
    }
    completionHandler()
}
Apple Watch
Apple Watch
<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Watch, WatchKit In-Depth, Part 1</td>
<td>Pacific Heights</td>
<td>Wednesday 9:00AM</td>
</tr>
<tr>
<td>Apple Watch, WatchKit In-Depth, Part 2</td>
<td>Pacific Heights</td>
<td>Wednesday 10:00AM</td>
</tr>
</tbody>
</table>
One More Thing...
One More Thing…

iOS 8 compatibility
Summary
Summary

iOS Notifications
Summary

iOS Notifications

- Silent Notifications
Summary

iOS Notifications
• Silent Notifications
• User Notifications
Summary

iOS Notifications

- Silent Notifications
- User Notifications
- Actions
Summary

iOS Notifications
• Silent Notifications
• User Notifications
• Actions
• Text input — New!
Summary

iOS Notifications
• Silent Notifications
• User Notifications
• Actions
• Text input — New!
• Apple Watch
Apple Push Notification Service

Gokul Thirumalai
Apple Push Notification Service
Apple Push Notification Service

Review

New Provider API
Flow

APNS

Provider

Client App
Flow

APNS

Provider

Client App
Flow

APNS

Provider

Client App

Flow diagram showing the relationship between APNS, Provider, and Client App.
Flow

APNS

Provider

Client App
Flow

APNS

Provider

Client App
Flow

APNS

Provider

Client App
Feedback

APNS Feedback

APNS

Provider
Feedback

APNS
Feedback

APNS

Provider
Feedback

APNS
Feedback

APNS

Provider

iPhone
Feedback

APNS Feedback

APNS

Provider
Large Device Tokens

Coming in 2016
New Provider API
HTTP/2
HTTP/2
HTTP/2

Request-response
HTTP/2

Request-response
Multiplexed
HTTP/2

Request-response
Multiplexed
Binary
Flow

APNS

Provider

Client App

Flow diagram showing the interaction between APNS, Provider, and Client App.
Flow

APNS → HTTP/2 → Provider

APNS → Client App

Provider → Client App
HTTP/2 Provider API

Connection

APNS <-> Provider
HTTP/2 Provider API

Connection

APNS

Provider
HTTP/2 Provider API

Connection

APNS

SETTINGS

Provider
HTTP/2 Provider API

Request

APNS → Provider
HTTP/2 Provider API

Request

POST
HTTP/2 Provider API

Request

POST...

APNS

POST...

Provider

POST
HTTP/2 Provider API

Response

APNS -> Provider
HTTP/2 Provider API

Response

APNS ----> 200 OK ----> Provider
HTTP/2 Provider API

Response

400 BAD REQUEST

{ 'reason' : ... }
HTTP/2 Provider API

Request example

HEADERS
- END_STREAM
+ END_HEADERS
:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
content-length = 41
apns-id = de305d54-75b4-431b-adb2-eb6b9e546014
apns-expiration = 0

DATA
+ END_STREAM
{ "aps" : { "alert" : "Hello HTTP/2" } }
HTTP/2 Provider API

Request example

HEADERS
- END_STREAM
+ END_HEADERS

:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
content-length = 41
apns-id = de305d54-75b4-431b-adb2-eb6b9e546014
apns-expiration = 0

DATA
+ END_STREAM

{ "aps" : { "alert" : "Hello HTTP/2" } }
HTTP/2 Provider API

Request example

HEADERS
- END_STREAM
+ END_HEADERS
:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
content-length = 41
apns-id = de305d54-75b4-431b-adb2-eb6b9e546014
apns-expiration = 0

DATA
+ END_STREAM
{ "aps" : { "alert" : "Hello HTTP/2" } }
HTTP/2 Provider API

Request example

HEADERS
- END_STREAM
+ END_HEADERS
:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
content-length = 41
apns-id = de305d54-75b4-431b-adb2-eb6b9e546014
apns-expiration = 0

DATA
+ END_STREAM
{ "aps" : { "alert" : "Hello HTTP/2" } }
HTTP/2 Provider API

Request example

HEADERS
- END_STREAM
+ END_HEADERS
:authority = api.push.apple.com
:method = POST
:path = /3/device/ad2bcd38f6773cdad050411...
content-length = 41
apns-id = de305d54-75b4-431b-adb2-eb6b9e546014
apns-expiration = 0

DATA
+ END_STREAM
{
  "aps": {
    "alert": "Hello HTTP/2"
  }
}
HTTP/2 Provider API

Response example

HEADERS
+ END_STREAM
+ END-Headers
:status = 200
HTTP/2 Provider API

Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 400
content-type = application/json
content-length = 31

DATA
+ END_STREAM
{
  "reason": "BadDeviceToken"
}
HTTP/2 Provider API
Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 400
content-type = application/json
content-length = 31

DATA
+ END_STREAM

{ "reason": "BadDeviceToken" }
HTTP/2 Provider API

Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 400
content-type = application/json
content-length = 31

DATA
+ END_STREAM
{ "reason": "BadDeviceToken" }
HTTP/2 Provider API

Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 400
content-type = application/json
content-length = 31

DATA
+ END_STREAM
{ "reason": "BadDeviceToken" }
Instant Feedback
Instant Feedback

![Diagram showing the relationship between APNS and Provider with APNS Feedback]
Instant Feedback

APNS → Provider
Instant Feedback

APNS

Request

Feedback in Response

Provider
Instant Feedback
Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 410
content-type = application/json
content-length = 48

DATA
+ END_STREAM
{
  "timestamp": 1432819882
}
Instant Feedback
Response example

HEADERS
- END_STREAM
+ END_HEADERS
:status = 410
content-type = application/json
content-length = 48

DATA
+ END_STREAM
{
  "timestamp": 1432819882
}
Simplified Certificate Handling
Simplified Certificate Handling
Simplified Certificate Handling

Application push
Simplified Certificate Handling

Application push
VOIP push
Simplified Certificate Handling

Application push
VOIP push
Watch Complication push
Simplified Certificate Handling

Application push
VOIP push
Watch Complication push
Development and production environments
Simplified Certificate Handling

Application push
VOIP push
Watch Complication push
Development and production environments
New Provider API

Review

HTTP/2 request-response

Instant feedback

Simplified certificate handling

4KB
More Information

Documentation

Technical Support
Apple Developer Forums
http://developer.apple.com/forums

Developer Technical Support
http://developer.apple.com/support/technical

General Inquiries
Paul Danbold, Core OS Evangelist
danbold@apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating Complications with ClockKit</td>
<td>Pacific Heights</td>
<td>Wednesday 11:00AM</td>
</tr>
<tr>
<td>Networking with NSURLSession</td>
<td>Pacific Heights</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>Introducing Watch Connectivity</td>
<td>Pacific Heights</td>
<td>Thursday 11:00AM</td>
</tr>
<tr>
<td>Related Labs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notifications Lab</td>
<td>Frameworks Lab E</td>
<td>Friday 2:30PM</td>
</tr>
</tbody>
</table>