Creating Extensions for iOS and OS X, Part 1
Extending Apps

Session 205
Ian Baird
iOS Frameworks Engineer

Matt Gamble
iOS Notifications Engineer

Guy Fullerton
iOS Social & Accounts Engineer

© 2014 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.
Agenda

Extensions and Extension Points
Agenda

Extensions and Extension Points
Today Extensions
Agenda

Extensions and Extension Points
Today Extensions
Share Extensions
What are Extensions?
Monday, June 2nd

Mostly sunny currently. The high will be 81°. Partly cloudy tonight with a low of 55°.

SportsCenter

<table>
<thead>
<tr>
<th>Team</th>
<th>Record</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athletics</td>
<td>30-17</td>
<td>5/23</td>
<td>4:07 PM</td>
</tr>
<tr>
<td>Blue Jays</td>
<td>26-22</td>
<td>OAK -124</td>
<td></td>
</tr>
<tr>
<td>Twins</td>
<td>23-21</td>
<td>5/23</td>
<td>7:15 PM</td>
</tr>
<tr>
<td>Giants</td>
<td>29-18</td>
<td>SF -154</td>
<td></td>
</tr>
</tbody>
</table>
What will your verse be?
Everyone has something to share.

AirDrop
Share instantly with people nearby. If they do not appear automatically, ask them to open Control Center and turn on AirDrop.

Message  Mail  Twitter  Facebook

Add Bookmark  Add to Reading List  Add to Home Screen  Annotate Image

Cancel
What will your verse be?
Everyone has something to share.

AirDrop
Share instantly with people nearby. If they do not appear automatically, ask them to open Control Center and turn on AirDrop.
Hi!

Weekend at the lake

From: Johnny Appleseed <jonnyappleseed@apple.com>
Miss you already

To: Stefan Schuch

Cc/Bcc: 

Subject: Miss you already

Paul and I can’t believe the week went by so quickly
Delivering Extensions
Delivering Extensions

Extension Container
Delivering Extensions

Extension Container
Delivering Extensions

Extension Container

[Diagram showing three puzzle pieces connected to an extension container]
What are Extension Points?

Binding
What are Extension Points?

Binding

NotificationCenter.framework

Today Widget
What are Extensions?

Purpose built binaries
What are Extensions?

Purpose built binaries
Not apps
What are Extensions?

- Purpose built binaries
- Not apps
- Accessed via Apple frameworks code
What are Extensions?

Purpose built binaries
Not apps
Accessed via Apple frameworks code
Not app to app IPC
What are Extensions?

Purpose built binaries
Not apps
Accessed via Apple frameworks code
Not app to app IPC
  • Still use – [UIApplication openURL:]
What are Extension Points?

Overview

Mark extensible parts of the system
What are Extension Points?

Overview

Mark extensible parts of the system
Packaged in System Frameworks
What are Extension Points?

Overview

Mark extensible parts of the system
Packaged in System Frameworks
Combine API and Policy
What are Extension Points?

Policy

Launch characteristics
What are Extension Points?

Policy

Launch characteristics
Presentation
What are Extension Points?

Policy

Launch characteristics

Presentation

View controller and non view
Activating Extensions

Extension Host

Extension
Activating Extensions

Extension Host

Extension
Directly Launch Extension Host Extension
Summary

Small pieces of functionality
Summary

Small pieces of functionality
Extend the system in new and interesting ways
Summary

Small pieces of functionality
Extend the system in new and interesting ways
Extension Points mark extendable parts of the system
Notification Center Extensions
The widgets you need and deserve

Matt Gamble
iOS Notifications Engineer
Notification Center Widgets

Widgets are view controllers
Notification Center Widgets

Widgets are view controllers

Lifecycle, containment, etc.
Notification Center Widgets

Widgets are view controllers

Lifecycle, containment, etc.

Appearance calls:

- (void)viewWillAppear:(BOOL)animated;
- (void)viewDidAppear:(BOOL)animated;
- (void)viewWillDisappear:(BOOL)animated;
- (void)viewDidDisappear:(BOOL)animated;
- (void)viewWillDisappear:(BOOL)animated;
- (void)viewDidDisappear:(BOOL)animated;
Notification Center Widgets

Widgets are view controllers

Lifecycle, containment, etc.

Appearance calls:
- (void)viewWillAppear:(BOOL)animated;
- (void)viewDidAppear:(BOOL)animated;
- (void)viewWillDisappear:(BOOL)animated;
- (void)viewDidDisappear:(BOOL)animated;
- (void)viewWillDisappear:(BOOL)animated;
- (void)viewDidDisappear:(BOOL)animated;
Notification Center Widgets

Performance
Notification Center Widgets

Performance

Load cached data for seamless transitions
Notification Center Widgets

Performance

Load cached data for seamless transitions
Kick off expensive operations early and in the background
Notification Center Widgets

Performance

- Load cached data for seamless transitions
- Kick off expensive operations early and in the background
- Cache fetched data (see “Load cached data for seamless transitions”)

Notification Center Widgets
Handling layout
Notification Center Widgets

Handling layout

The Notification Center sets the frames
The Notification Center sets the frames
Adjusting Height
Notification Center Widgets
Handling layout

The Notification Center sets the frames

Adjusting Height
1. Auto layout
Notification Center Widgets

Handling layout

The Notification Center sets the frames

Adjusting Height

1. Auto layout
2. setPreferredContentSize:
Notification Center Widgets
Handling layout

The Notification Center sets the frames

Adjusting Height
1. Auto layout
2. `setPreferredContentSize`:

Same across iOS and OS X
Notification Center Widgets
Handling layout
Notification Center Widgets

Handling layout

- (IBAction)action:(id)sender {

    CGSize updatedSize = [self preferredContentSize];
    updatedSize.height = 100.0;

    [self setPreferredContentSize:updatedSize];

}
- (IBAction)action:(id)sender {

    CGSize updatedSize = [self preferredContentSize];
    updatedSize.height = 100.0;

    [self setPreferredContentSize:updatedSize];
}

Notification Center Widgets
Handling layout
Notification Center Widgets

Animating along with height changes
Notification Center Widgets

Animating along with height changes

- (void)viewWillTransitionToSize:(CGSize)size
  withTransitionCoordinator:(id <UIViewControllerTransitionCoordinator>)coordinator {

    [coordinator animateAlongsideTransition:

        ^(id<UIViewControllerTransitionCoordinatorContext> context) {
            [view setAlpha:0.0];
        }

        completion:

        ^(id<UIViewControllerTransitionCoordinatorContext> context) {
            [view removeFromSuperview];
        }
    ];

}
Notification Center Widgets

Animating along with height changes

- (void)viewWillTransitionToSize:(CGSize)size withTransitionCoordinator:(id <UIViewControllerTransitionCoordinator>)coordinator {

    [coordinator animateAlongsideTransition:
     ^(id<UIViewControllerTransitionCoordinatorContext> context) {
        [view setAlpha:0.0];
    }

    completion:
     ^(id<UIViewControllerTransitionCoordinatorContext> context) {
        [view removeFromSuperview];
    }]

}
Notification Center Widgets

Animating along with height changes

- (void)viewWillTransitionToSize:(CGSize)size
  withTransitionCoordinator:(id <UIViewControllerTransitionCoordinator>)coordinator {

  [coordinator animateAlongsideTransition:
    ^(id<UIViewControllerTransitionCoordinatorContext> context) {
      [view setAlpha:0.0];
    }

    completion:
      ^(id<UIViewControllerTransitionCoordinatorContext> context) {
        [view removeFromSuperview];
      }
    ];

}
Notification Center Widgets

Animating along with height changes

- (void)viewWillTransitionToSize:(NSSize)newSize {

    [NSAnimationContext runAnimationGroup:^(NSAnimationContext *context) {

        [view setAlpha:0.0];

    } completionHandler:^{

        [view removeFromSuperview];

    }];

}
- (void)viewWillTransitionToSize:(NSSize)newSize {

    [NSAnimationContext runAnimationGroup:^{(NSAnimationContext *context) {

        [view setAlpha:0.0];

    } completionHandler:^{

        [view removeFromSuperview];

    }}];

}
- (void)viewWillTransitionToSize:(NSSize)newSize {

    [NSAnimationContext runAnimationGroup:^(NSAnimationContext *context) {

        [view setAlpha:0.0];

    } completionHandler:^(NSAnimationContext *context) {

        [view removeFromSuperview];

    }];

}
Notification Center Widgets
NCWidgetProviding
Notification Center Widgets
NCWidgetProviding

-(void)widgetPerformUpdateWithCompletionHandler:(void (^)(NCUpdateResult result))completionHandler {

    [self performUpdate];

    NCUpdateResult updateResult;

    if (/* Hey look, I have new content! */) {
        updateResult = NCUpdateResultNewData;
    } else if (/* Hm, nothing new here. */) {
        updateResult = NCUpdateResultNoData;
    } else if (/* Uh-oh... */) {
        updateResult = NCUpdateResultFailed;
    }

    completionHandler(updateResult);
}
Notification Center Widgets
NCWidgetProviding

-(void)widgetPerformUpdateWithCompletionHandler:(void (^)(NCUpdateResult result))completionHandler {

    [self performUpdate];

    NCUpdateResult updateResult;

    if (/* Hey look, I have new content! */) {
        updateResult = NCUpdateResultNewData;
    } else if ( /* Hm, nothing new here. */) {
        updateResult = NCUpdateResultNoData;
    } else if ( /* Uh-oh... */) {
        updateResult = NCUpdateResultFailed;
    }

    completionHandler(updateResult);
}

Notification Center Widgets
NCWidgetProviding

-(void)widgetPerformUpdateWithCompletionHandler:(void (^)(NCUpdateResult result))completionHandler {

    [self performUpdate];

    NCUpdateResult updateResult;

    if (/* Hey look, I have new content! */) {
        updateResult = NCUpdateResultNewData;
    } else if ( /* Hm, nothing new here. */) {
        updateResult = NCUpdateResultNoData;
    } else if ( /* Uh-oh... */) {
        updateResult = NCUpdateResultFailed;
    }

    completionHandler(updateResult);
}
- (void)widgetPerformUpdateWithCompletionHandler:(void (^)(NCUpdateResult result))completionHandler {

  [self performUpdate];

  NCUpdateResult updateResult;

  if (/* Hey look, I have new content! */) {
      updateResult = NCUpdateResultNewData;
  } else if ( /* Hm, nothing new here. */) {
      updateResult = NCUpdateResultNoData;
  } else if ( /* Uh-oh… */) {
      updateResult = NCUpdateResultFailed;
  }

  completionHandler(updateResult);
}
Notification Center Widgets

NCWidgetProviding

-(void)widgetPerformUpdateWithCompletionHandler:(void (^)(NCUpdateResult result))completionHandler {

    [self performUpdate];

    NCUpdateResult updateResult;

    if (/* Hey look, I have new content! */) {
        updateResult = NCUpdateResultNewData;
    } else if (/* Hm, nothing new here. */) {
        updateResult = NCUpdateResultNoData;
    } else if (/* Uh-oh... */) {
        updateResult = NCUpdateResultFailed;
    }

    completionHandler(updateResult);
}

Notification Center Widgets
NCWidgetProviding

- (void)widgetPerformUpdateWithCompletionHandler:(void (^)(NCUpdateResult result))completionHandler {

    [self performUpdate];

    NCUpdateResult updateResult;

    if (/* Hey look, I have new content! */) {
        updateResult = NCUpdateResultNewData;
    } else if ( /* Hm, nothing new here. */) {
        updateResult = NCUpdateResultNoData;
    } else if ( /* Uh-oh... */) {
        updateResult = NCUpdateResultFailed;
    }

    completionHandler(updateResult);
}
Demo
Clowning around
Notification Center Widgets

Summary
Notification Center Widgets

Summary

Widgets are view controllers
Notification Center Widgets

Summary

Widgets are view controllers
Widgets should resume immediately
Notification Center Widgets

Summary

Widgets are view controllers
Widgets should resume immediately
Notification Center handles layout
• Express preferred height with constraints or preferredContentSize
Notification Center Widgets

Summary

Widgets are view controllers
Widgets should resume immediately
Notification Center handles layout
• Express preferred height with constraints or preferredContentSize
Animate content along with resize animation
Widgets are view controllers
Widgets should resume immediately
Notification Center handles layout
  • Express preferred height with constraints or `preferredContentSize`
Animate content along with resize animation
Handle update requests
Share Extensions

Guy Fullerton
iOS Social & Accounts Engineer
Share Extensions
Share Extensions
AirDrop
Share instantly with people nearby. If they do not appear automatically, ask them to open Control Center and turn on AirDrop.
Great day for kites!
Share Extensions

Display name
Activation rules
Implement a ViewController
Upload the content
Signal completion/cancellation
Display Name
Display Name

CFBundleDisplayName
Activation Rules
Activation Rules

extensionContext

image 1
image 2
video
Activation Rules

extensionContext

image 1  image 2  video

Social Network Extension

Photo Sharing Extension

Microblog Extension

Video Sharing Extension
Activation Rules

extensionContext

image 1  image 2  video

Social Network Extension
Photo Sharing Extension
Microblog Extension
Video Sharing Extension
## Activation Rules

### Predicate

<table>
<thead>
<tr>
<th>▼ NSExtension</th>
<th>▲</th>
<th>Dictionary</th>
<th>(4 items)</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼ NSExtensionAttributes</td>
<td>Dictionary</td>
<td>(1 item)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>NSExtensionActivationRule</strong></td>
<td>String</td>
<td>SUBQUERY(extensionItems, $extensionItem, SUBQUERY($extensionItem))</td>
</tr>
</tbody>
</table>
Activation Rules

Predicate

```
SUBQUERY(extensionItems, $extensionItem, 
SUBQUERY($extensionItem.attachments, $attachment, ANY 
$attachment.registeredTypeIdentifiers UTI-CONFORMS-TO "public.image").@count == 1).@count == 1 OR SUBQUERY(extensionItems, $extensionItem, 
SUBQUERY($extensionItem.attachments, $attachment, ANY 
$attachment.registeredTypeIdentifiers UTI-CONFORMS-TO "com.adobe.pdf").@count == 1).@count == 1
```
# Activation Rules

## Condensed rules

<table>
<thead>
<tr>
<th>NSExtension</th>
<th>Dictionary</th>
<th>(4 items)</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼ NSExtensionAttributes</td>
<td>Dictionary</td>
<td>(1 item)</td>
</tr>
<tr>
<td>▼ NSExtensionActivationRule</td>
<td>Dictionary</td>
<td>(6 items)</td>
</tr>
<tr>
<td>NSExtensionActivationSupportsText</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>NSExtensionActivationSupportsImageWithMaxCount</td>
<td>Number</td>
<td>10</td>
</tr>
<tr>
<td>NSExtensionActivationSupportsVideoWithMaxCount</td>
<td>Number</td>
<td>1</td>
</tr>
<tr>
<td>NSExtensionActivationSupportsFileWithMaxCount</td>
<td>Number</td>
<td>0</td>
</tr>
<tr>
<td>NSExtensionActivationSupportsWebURLWithMaxCount</td>
<td>Number</td>
<td>1</td>
</tr>
<tr>
<td>NSExtensionActivationSupportsWebPageWithMaxCount</td>
<td>Number</td>
<td>0</td>
</tr>
</tbody>
</table>
Activation Rules

Web content

NSExtensionActivationSupportsWebURLWithMaxCount

NSExtensionActivationSupportsWebPageWithMaxCount
Implement Your ViewController
subclass UI/NSViewController
Implement Your ViewController
subclass SLComposeServiceViewController
- (void)didSelectPost {
    NSString *confName = @"com.example.photouploader.backgroundconfiguration";
    NSURLSessionConfiguration *conf =
        [NSURLSessionConfiguration backgroundSessionConfiguration:confName];

    NSURLSession *session = [NSURLSession sessionWithURLConfiguration:conf
deflegate:self delegateQueue:nil];

    NSURLRequest *request = [self requestForExtensionItems];
    NSURLSessionUploadTask *upload = [session uploadTaskWithURLStreamedRequest:request];
    [upload resume];

    [self.extensionContext completeRequestReturningItems:nil completionHandler:nil];
}
- (void)didSelectPost {
    NSString *confName = @"com.example.photouploader.backgroundconfiguration";
    NSURLSessionConfiguration *conf =
        [NSURLSessionConfiguration backgroundSessionConfiguration:confName];

    URLSession *session = [NSURLSession sessionWithURLSessionConfiguration:conf
delegate:self delegateQueue:nil];

    NSURLRequest *request = [self requestForExtensionItems];
    NSURLSessionUploadTask *upload = [session uploadTaskWithURLRequest:request];
    [upload resume];

    [self.extensionContext completeRequestReturningItems:nil completionHandler:nil];
}
- (void)didSelectPost {
    NSString *confName = @“com.example.photouploader.backgroundconfiguration”; 
    NSURLSessionConfiguration *conf = 
        [NSURLSessionConfiguration backgroundSessionConfiguration:confName];

    NSURLSession *session = [NSURLSession sessionWithURLConfiguration:conf 
    delegate:self delegateQueue:nil];

    NSURLRequest *request = [self requestForExtensionItems];
    NSURLSessionUploadTask *upload = [session uploadTaskWithURL:request];
    [upload resume];

    [self.extensionContext completeRequestReturningItems:nil completionHandler:nil];}

- (void)didSelectPost {
    NSString *confName = @“com.example.photoblog.backgroundconfiguration”;
    NSURLSessionConfiguration *conf =
        [NSURLSessionConfiguration backgroundSessionConfiguration:confName];

    NSURLSession *session = [NSURLSession sessionWithConfiguration:conf
delegate:self delegateQueue:nil];

    NSURLRequest *request = [self requestForExtensionItems];
    NSURLSessionUploadTask *upload = [session uploadTaskWithStreamedRequest:request];
    [upload resume];

    [self.extensionContext completeRequestReturningItems:nil completionHandler:nil];
}
- (void)didSelectCancel {
    NSError *error = [NSError errorWithDomain:NSCocoaErrorDomain
        code:NSUserCancelledError userInfonil];
    [self.extensionContext cancelRequestWithError:error];
}
Cancellation

- (void)didSelectCancel {
    NSError *error = [NSError errorWithDomain:NSCocoaErrorDomain
code:NSUserCancelledError userInfo:nil];
    [self.extensionContext cancelRequestWithError:error];
}
Performance

Display your Share UI quickly!

SLComposeServiceViewController

-presentationAnimationDidFinish
Demo
Photo blog extension
Share Extensions

SLComposeServiceViewController for standard UI
UI/NSViewController for custom UI

Activation rules
Use Background NSURLSession
Summary

Extend both iOS and OS X
Today Widgets
Share extensions
More Information

Jake Behrens
App Frameworks Evangelist
behrens@apple.com

Documentation
App Extension Programming Guide
http://apple.com

Apple Developer Forums
http://devforums.apple.com
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating Extensions, Part 2</td>
<td>Mission</td>
<td>Wednesday 11:30AM</td>
</tr>
</tbody>
</table>

## Labs

<table>
<thead>
<tr>
<th>Lab</th>
<th>Frameworks</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensions Lab</td>
<td>Lab A</td>
<td>Tuesday 3:15PM</td>
</tr>
<tr>
<td>Extensions Lab</td>
<td>Lab B</td>
<td>Thursday 2:00PM</td>
</tr>
</tbody>
</table>