Window Management in Your Multitasking App
Activate, refresh, and destroy windows

Giovanni Tarducci, iOS System UI
Programmatic Session Management
On behalf of your users

Activate
Refresh
Destroy
Demo
Programmatic Session Management
On behalf of your users

Activate

Refresh

Destroy
Programmatic Session Management
On behalf of your users

Activate
Refresh
Destroy
Activate a session only in response to direct and local user interaction
Session Activation

New method on `UIApplication` to activate

- New scene session
- Existing one

```swift
open func requestSceneSessionActivation(_ sceneSession: UISceneSession?,
userActivity: NSUserActivity?,
options: UIScene.ActivationRequestOptions?,
errorHandler: ((Error) -> Void)? = nil)
```
Session Activation
New scene session
Session Activation
New scene session

UIApplication

Map Scene Session

Window Scene

Window

Root View Controller
class AppDelegate: UIResponder, UIApplicationDelegate {

    func activateSessionForClown(_ clown: Clown!,
                     sourceScene: UIWindowScene?,
                     errorHandler: ((Error) -> Void)? = nil)
    {

        let existingSession = existingSessionForClown(clown)
        let userActivity = userActivityForClown(clown)
        let options =UIScene.ActivationRequestOptions.init()
        options.requestingScene = sourceScene

        UIApplication.shared.requestSceneSessionActivation(existingSession,
                                                           userActivity: userActivity,
                                                           options: options,
                                                           errorHandler: errorHandler)
    }
}
// ClownTown: Session Activation

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func activateSessionForClown(_ clown: Clown!,
        sourceScene: UIWindowScene?,
        errorHandler: ((Error) -> Void)? = nil)
    {
        let existingSession = existingSessionForClown(clown)
        let userActivity = userActivityForClown(clown)
        let options = UIScene.ActivationRequestOptions.init()  
        options.requestingScene = sourceScene
        UIApplication.shared.requestSceneSessionActivation(existingSession,
            userActivity: userActivity,  
            options: options,  
            errorHandler: errorHandler)  
    }
}
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func activateSessionForClown(_ clown: Clown!,
        sourceScene: UIWindowScene?,
        errorHandler: ((Error) -> Void)? = nil)
    {
        let existingSession = existingSessionForClown(clown)
        let userActivity = userActivityForClown(clown)

        let options = UIScene.ActivationRequestOptions.init()
        options.requestingScene = sourceScene
        UIApplication.shared.requestSceneSessionActivation(existingSession,
            userActivity: userActivity,
            options: options,
            errorHandler: errorHandler)
    }
}
// ClownTown: Session Activation

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func activateSessionForClown(_ clown: Clown!,
        sourceScene: UIWindowScene?,
        errorHandler: ((Error) -> Void)? = nil)
    {
        let existingSession = existingSessionForClown(clown)
        let userActivity = userActivityForClown(clown)

        let options = UIScene.ActivationRequestOptions.init()
        options.requestingScene = sourceScene

        UIApplication.shared.requestSceneSessionActivation(existingSession,
            userActivity: userActivity,
            options: options,
            errorHandler: errorHandler)
    }
}
// ClownTown: Session Activation

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func activateSessionForClown(_ clown: Clown!,
        sourceScene: UIWindowScene?,
        errorHandler: ((Error) -> Void)? = nil)
    {
        let existingSession = existingSessionForClown(clown)
        let userActivity = userActivityForClown(clown)
        let options = UIScene.ActivationRequestOptions.init()
        options.requestingScene = sourceScene
        UIApplication.shared.requestSceneSessionActivation(existingSession,
            userActivity: userActivity,
            options: options,
            errorHandler: errorHandler)
    }
}
// ClownTown: Session Activation

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
    func activateSessionForClown(_ clown: Clown!,
        sourceScene: UIWindowScene?,
        errorHandler: ((Error) -> Void)? = nil)
    {
        let existingSession = existingSessionForClown(clown)
        let userActivity = userActivityForClown(clown)
        let options =UIScene.ActivationRequestOptions.init()
        options.requestingScene = sourceScene
        UIApplication.shared.requestSceneSessionActivation(existingSession,
            userActivity: userActivity,
            options: options,
            errorHandler: errorHandler)
    }
}
Session Activation

New scene session
Session Activation
New scene session.

UIApplication
Map Scene Session
Window Scene
Window
Root View Controller
Session Activation

New scene session

UIApplication

Map Scene Session
- Window Scene
  - Window
    - Root View Controller

Detail Scene Session
- Window Scene
  - Window
    - Root View Controller
Session Activation
New scene session

- UIApplication
  - Map Scene Session
    - Window Scene
      - Window
        - Root View Controller
    - Detail Scene Session
      - Window Scene
        - Window
        - Root View Controller
Session Activation

New scene session

UIApplication

Map Scene Session

Window Scene

Window

Root View Controller

Detail Scene Session

Window Scene

Window

Root View Controller
Session Activation
New scene session

UIApplication

Map Scene Session
- Window Scene
- Window
- Root View Controller

Detail Scene Session
- Window Scene
- Window
- Root View Controller

requestSceneSessionActivation
Session Activation

New scene session

UIApplication

Map Scene Session ➔ Scene Session

Window Scene ➔ Window

Window ➔ Root View Controller

UIApplication

requestSceneSessionActivation
Session Activation
New scene session

UIApplication

Map Scene Session
Scene Session
Window Scene
Window
Root View Controller

UIApplication
requestSceneSessionActivation

AppDelegate
configurationForSession
Session Activation
New scene session

UIApplication

Map Scene Session
Scene Session

Window Scene
Window

Root View Controller

UIApplication
requestSceneSessionActivation

App Delegate
configurationForSession
Session Activation

New scene session

UIApplication

Map Scene Session

Window Scene

Window

Root View Controller

Detail Scene Session

Window Scene

Window

Root View Controller

UIApplication

requestSceneSessionActivation

AppDelegate

configurationForSession
Session Activation
New scene session

UIApplication

Map Scene Session

Detail Scene Session

Window Scene

Window

Root View Controller

UIApplication

requestSceneSessionActivation

AppDelegate

configurationForSession
Session Activation

New scene session

UIApplication

Map Scene Session
  Window Scene
    Window
      Root View Controller

Detail Scene Session
  Window Scene
    Window
      Root View Controller

UIApplication
  requestSceneSessionActivation
    AppDelegate
      configurationForSession
Session Activation
New scene session

UIApplication
- requestSceneSessionActivation

AppDelegate
- configurationForSession

Scene Delegate
- sceneWillConnectToSession:
Session Activation

New scene session

UIApplication

Map Scene Session
Detail Scene Session

Window Scene

Window

Root View Controller

App Delegate

configurationForSession

Scene Delegate

sceneWillConnectToSession:
Session Activation

**Existing session**

- UIApplication
  - Map Scene Session
    - Window Scene
      - Window
        - Root View Controller
  - Detail Scene Session
    - Window Scene
      - Window
        - Root View Controller

-UIApplication
  - requestSceneSessionActivation

- AppDelegate
  - configurationForSession

- Scene Delegate
  - sceneWillConnectToSession:
Session Activation

Existing session

UIApplication

Map Scene Session

Window Scene

Window

Root View Controller

Detail Scene Session

Window Scene

Window

Root View Controller

Scene Delegate

sceneWillConnectToSession:
Session Activation

Existing session

UIApplication

Map Scene Session

Detail Scene Session

Window Scene

Window Scene

Window

Window

Root View Controller

Root View Controller

Scene Delegate

sceneWillConnectToSession:

or

continueUserActivity:
Session Activation Summary

Call in response to direct and local user interaction
  • New or existing session

Implement App and Scene delegates
  • Select a configuration
  • Configure your window
Programmatic Session Management
On behalf of your users

Activate
Refresh
Destroy
Programmatic Session Management
On behalf of your users

Activate

Refresh

Destroy
Refresh a session for user-relevant updates
Session Refresh

Updates to shared data

New user-relevant data

Session and scene hygiene
Session Refresh

New method on UIApplication to refresh your scene session
• Acts on non-foreground scenes only

open func requestSceneSessionRefresh(_ sceneSession: UISceneSession)
Session Refresh

State restoration user activity

Activation conditions

Session UI and switcher snapshot
Model Updates

View controller

Long-lived Object

Related session has connected scene?

Yes

End

No

Refresh API call

Scene Foreground?

Yes

End

No

Snapshot

Refresh API call

Backgrounds New Scene + View Controller Layout + Snapshot
Session Refresh Summary

For user-relevant state updates only

Optimize your layout time

The system may fulfill the request at a later point in time
Programmatic Session Management
On behalf of your users

Activate
Refresh
Destroy
Programmatic Session Management
On behalf of your users

Activate
Refresh
Destroy
Destroy a session only in response to direct user interaction
New method on `UIApplication` to destroy a session

```swift
open func requestSceneSessionDestruction(_ sceneSession:UISceneSession,
    options:UIScene.DestructionRequestOptions?,
    errorHandler:((Error)->Void)? = nil)
```
extension UIWindowScene {

    @available(iOS 13.0, *)
    public enum DismissalAnimation : Int {
        case standard
        case commit
        case decline
    }

    @available(iOS 13.0, *)
    open class DestructionRequestOptions : UIScene.DestructionRequestOptions {
        open var windowDismissalAnimation: UIWindowScene.DismissalAnimation
    }
}
// UIScene DestructionRequestOptions

extension UIWindowScene {

@available(iOS 13.0, *)
public enum DismissalAnimation : Int {
    case standard
    case commit
    case decline
}

@available(iOS 13.0, *)
open class DestructionRequestOptions : UIScene.DestructionRequestOptions {
    open var windowDismissalAnimation: UIWindowScene.D dismissalAnimation
}
}
extension UIWindowScene {

    @available(iOS 13.0, *)
    public enum DismissalAnimation : Int {
        case standard
        case commit
        case decline
    }

    @available(iOS 13.0, *)
    open class DestructionRequestOptions : UIScene.DestructionRequestOptions {
        open var windowDismissalAnimation: UIWindowScene.DismissalAnimation
    }
}
Use the animation to acknowledge the user’s intent
Session Destruction Animations

Standard
Session Destruction Animations

Standard
Session Destruction Animations

Commit
Session Destruction Animations
Commit
Session Destruction Animations

Decline
Session Destruction Animations

Decline

Australia Trip
May 22, 2019 at 3:01 AM

Hi John,

Our family vacation to Australia was incredible. We visited Sydney, Brisbane, and Melbourne and had a lot of fun in each city. The weather was absolutely perfect. Even when we traveled inland, it never got too hot. And when we were on the ocean, we looked out and with perfect beach weather. In Sydney, we visited the Opera House and managed to get tickets to the Symphony Orchestra. We captured some great photos on our trip. Check them out:

[Picture of Sydney Opera House]
[Picture of a beach in Australia]

Hi Everyone,

Parking can be tough around the restaurant so I would arrive early to give yourself some time to find a spot. The restaurant’s address is below. Call if anything comes up. See you at 7:30!

Frances
3870 17th Street
San Francisco, CA 94114

Sent from my iPad
Session Destruction Summary

Call for direct or remote user interaction

Destroy session dedicated to user-deleted items

Pick the animation type to acknowledge user's intent
Empower your user’s interactions

Make your new features discoverable, powerful and easy-to-use

Use responsibly and have fun!
More Information

developer.apple.com/wwdc19/246