Deep Linking in tvOS

Session 246

Paul Schneider, tvOS Engineer
What is a Deep Link?

A URL referring to content that your application can display.

The content is often several steps away from your app’s main screen ("deep" in the navigation hierarchy).

The link jumps the user directly to the content.
Where Do Deep Links Come From?

Your top shelf extension is a likely source of deep link requests.

The same logic can be used to handle link requests from any application.
Use Universal Links

Allows parity with your iOS apps

Ensures that the link is routed to your application

Ensures that your app is expecting to handle the URL

Handling Deep Links Gracefully

Configure the ViewController hierarchy to cleanly display the URL content
Present the UI without flashes or transitions
Make sure that Menu takes the user to a predictable place
Two Top Shelf Item URLs

The displayURL is opened with the Select button
• Respond by presenting a detail page for the item

The playURL is opened with the Play button
• Respond by starting video playback
Recommended displayURL Handling

Opening a displayURL should show a detail page

Pressing Menu should return to the app’s initial screen

Pressing Menu again should return to the home screen
What's new in tvOS
Recommended playURL Handling

Opening a playURL should begin playing the video

Video should fade in from black

Pressing Menu should return to the video's detail page

Pressing Menu again should return the app's initial screen

Pressing Menu a third time should return to the home screen
What's new in tvOS

Play
Demo
func application(_ application: UIApplication, open url: URL, options: [UIApplicationOpenURLOptionsKey: Any] = [:]) -> Bool {
    if let (session, action) = Session.sessionFromURL(url) {
        UIView.performWithoutAnimation {

            let navController = self.window?.rootViewController as! UINavigationController
            navController.popToRootViewController(animated: false)

            let topVC = navController.topViewController
            topVC?.performSegue(withIdentifier: "showDetail", sender: session)

            if action == "play" {
                let detailVC = navController.topViewController
                detailVC?.performSegue(withIdentifier: "showPlayer", sender: nil)
            }
        }
    }
}
func application(_ application: UIApplication, open url: URL,
            options: [UIApplicationOpenURLOptionsKey: Any] = [:]) -> Bool {

    if let (session, action) = Session.sessionFromURL(url) {
        UIView.performWithoutAnimation {

            let navController = self.window?.rootViewController as! UINavigationController
            navController.popToRootViewController(animated: false)

            let topVC = navController.topViewController
            topVC?.performSegue(withIdentifier: "showDetail", sender: session)

            if action == "play" {
                let detailVC = navController.topViewController
                detailVC?.performSegue(withIdentifier: "showPlayer", sender: nil)
            }
        }
    }
}
func application(_ application: UIApplication, open url: URL, options: [UIApplicationOpenURLOptionsKey: Any] = [:]) -> Bool {
    if let (session, action) = Session.sessionFromURL(url) {
        UIView.performWithoutAnimation {

            let navController = self.window?.rootViewController as! UINavigationController
            navController.popToRootViewController(animated: false)

            let topVC = navController.topViewController
            topVC?.performSegue(withIdentifier: "showDetail", sender: session)

            if action == "play" {
                let detailVC = navController.topViewController
                detailVC?.performSegue(withIdentifier: "showPlayer", sender: nil)
            }
        }
    }
}
func application(_ application: UIApplication, open url: URL, options: [UIApplicationOpenURLOptionsKey: Any] = [:]) -> Bool {

    if let (session, action) = Session.sessionFromURL(url) {
        UIView.performWithoutAnimation {

            let navController = self.window?.rootViewController as! UINavigationController
            navController.popToRootViewController(animated: false)

            let topVC = navController.topViewController
            topVC?.performSegue(withIdentifier: "showDetail", sender: session)

            if action == "play" {
                let detailVC = navController.topViewController
                detailVC?.performSegue(withIdentifier: "showPlayer", sender: nil)
            }
        }
    }
}

func application(_ application: UIApplication, open url: URL, options: [UIApplicationOpenURLOptionsKey: Any] = [:]) -> Bool {

    // Check if the session can be extracted from the URL
    if let (session, action) = Session.sessionFromURL(url) {
        UIView.performWithoutAnimation {

            // Get the navigation controller
            let navController = self.window?.rootViewController as! UINavigationController
            navController.popToRootViewControllerAnimated(animated: false)

            // Get the top view controller
            let topVC = navController.topViewController
            topVC?.performSegue(withIdentifier: "showDetail", sender: session)

            // If 'play' action is specified, perform segue to show the player
            if action == "play" {
                let detailVC = navController.topViewController
                detailVC?.performSegue(withIdentifier: "showPlayer", sender: nil)
            }
        }
    }
}
Summary

Take the user to the linked content immediately

Give the user a predictable way to get back
More Information

https://developer.apple.com/wwdc17/246