

Settings

Mobile Application Development in iOS

School of EECS

Washington State University

Instructor: Larry Holder

Outline

- In-app settings
 - UserDefaults
- Device settings
 - Settings Bundle



Local Settings: UserDefaults

- Persistent key-value store for app
 - Accessed via singleton class instance: `UserDefaults.standard`
- Setting values
 - `UserDefaults.standard.set(value: <type>, forKey: String)`
 - `<type>`: `Bool`, `Int`, `Float`, `Double`, `String`, `URL`, `Date`
 - Or Array or Dictionary of these

```
let userName: String = "larry" // safer to force the type
let sortItems: Bool = true
let itemLimit: Int = 10
UserDefaults.standard.set(userName, forKey: "userName")
UserDefaults.standard.set(sortItems, forKey: "sortItems")
UserDefaults.standard.set(itemLimit, forKey: "itemLimit")
```

User Defaults

- Getting values
 - `UserDefaults.standard.<type>(forKey: String)`
 - `<type>`: bool, integer, float, double, string, url, dictionary, array, object
 - Returns `nil` if not set, except for bool, integer, float, double
- Removing values
 - `UserDefaults.standard.removeObject(forKey: String)`

```
// Usually in AppDelegate didFinishLaunchingWithOptions
if (UserDefaults.standard.object(forKey: "sortItems") == nil) {
    UserDefaults.standard.set(true, forKey: "sortItems")
}
// Later...
let sortItems = UserDefaults.standard.bool(forKey: "sortItems")
```

Sidebar: Singleton Class

- Create one, and only one, instance of a class

```
class MyGlobals {  
    static let sharedInstance = MyGlobals()  
  
    // Properties and methods  
    var userName = ""  
    var sortItems = true  
    var itemLimit = 10  
  
    // Prevent calling init() directly on class  
    private init() {}  
}  
  
// Access anywhere  
MyGlobals.sharedInstance.userName = "larry"
```

Singleton Struct for Constants

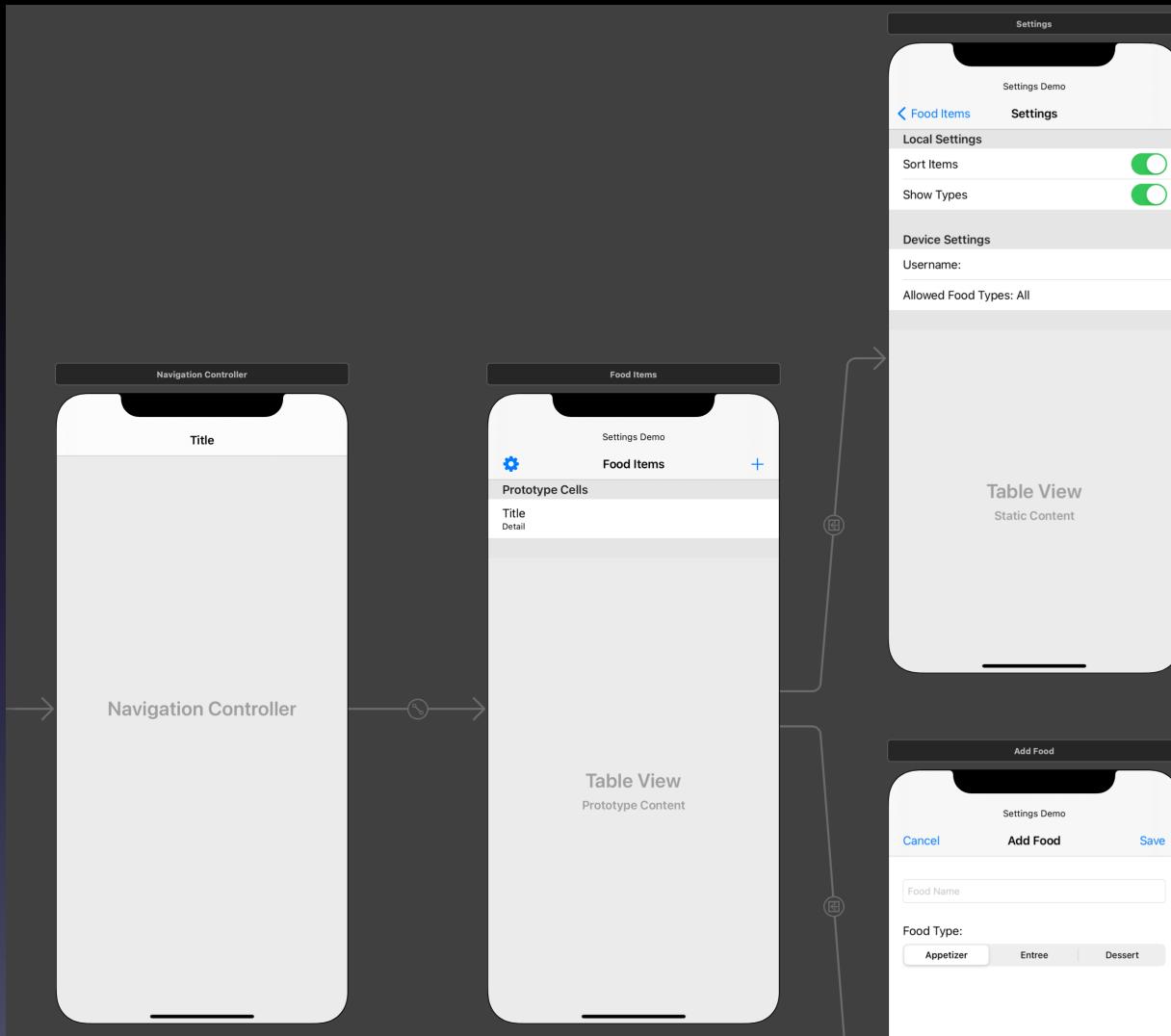
- Avoid misspelling key names

```
struct Constants {
    static let main = Constants()
    private init() {}

    let sortItemsKey = "sortItems"
    let showTypesKey = "showTypes"
    let userNameKey = "userName"
    let allowedTypesKey = "allowedTypes"
}

let userName = UserDefaults.standard.string(forKey:
    Constants.main.userNameKey) // auto-complete to the rescue
```

Settings Demo



Local Settings

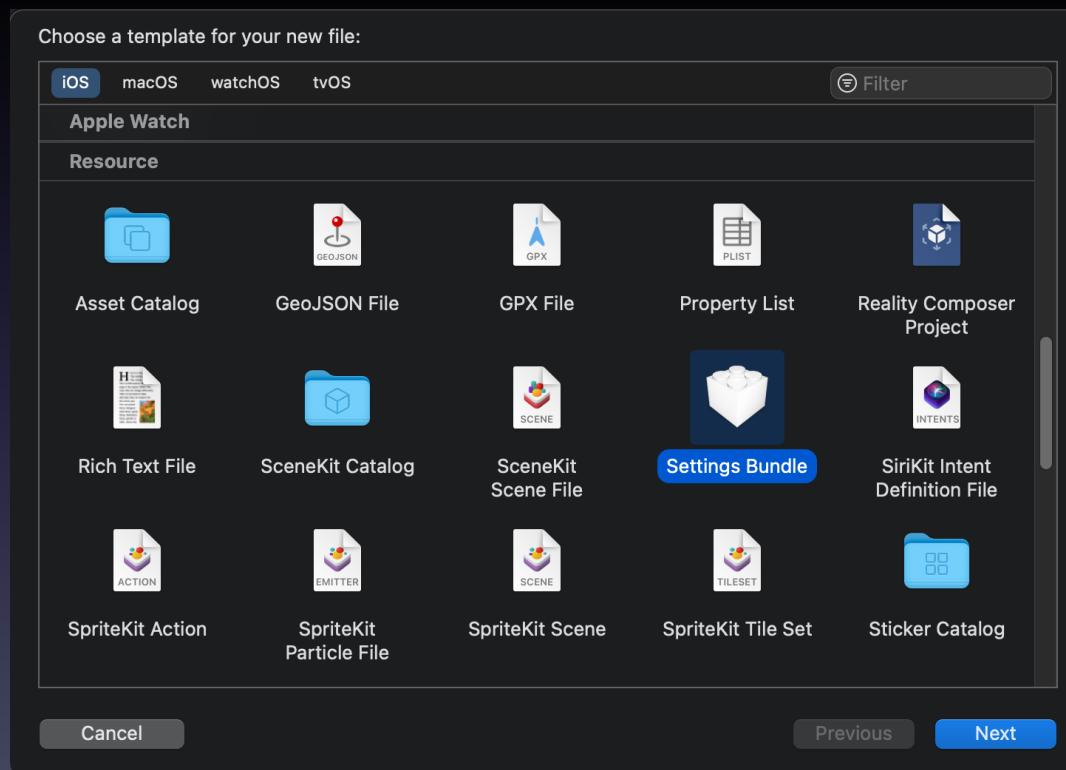
- How to know when settings changed?
 - Use `viewWillAppear` method in main view

```
override func viewWillAppear(_ animated: Bool) {  
    // Check settings and update view accordingly  
}
```

- Or, send notification to app when settings changed (next time...)

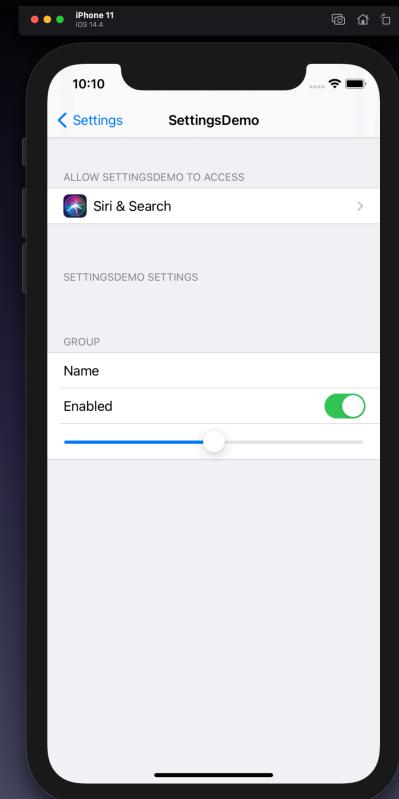
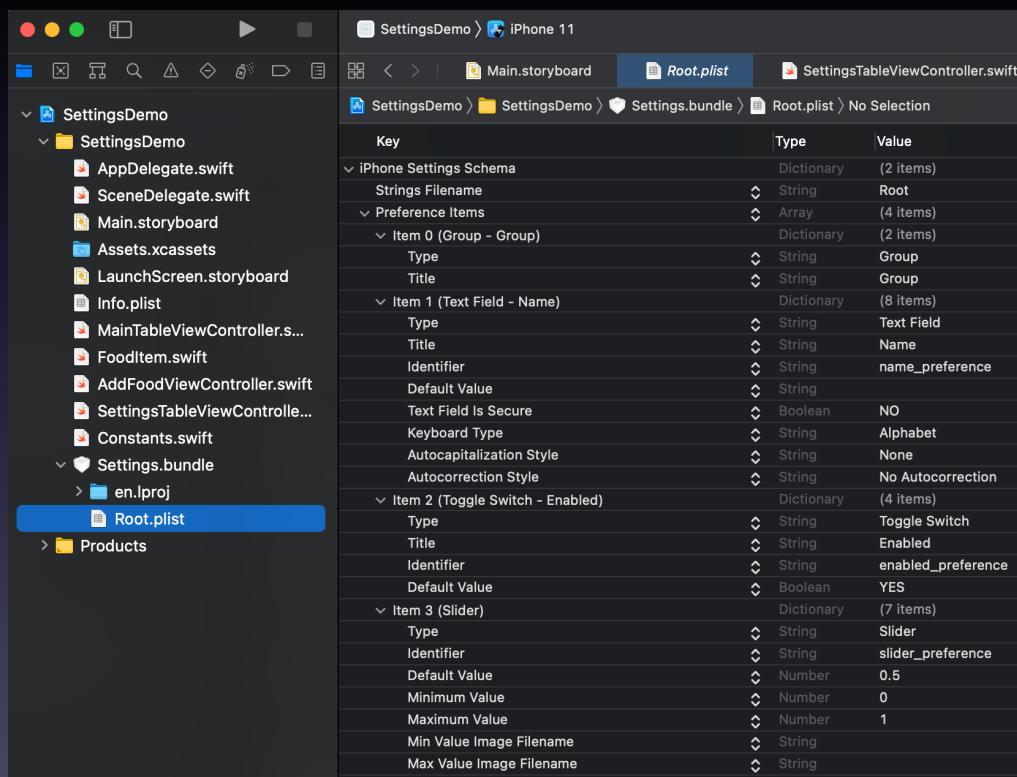
Device Settings: Settings Bundle

- Add Settings Bundle to app

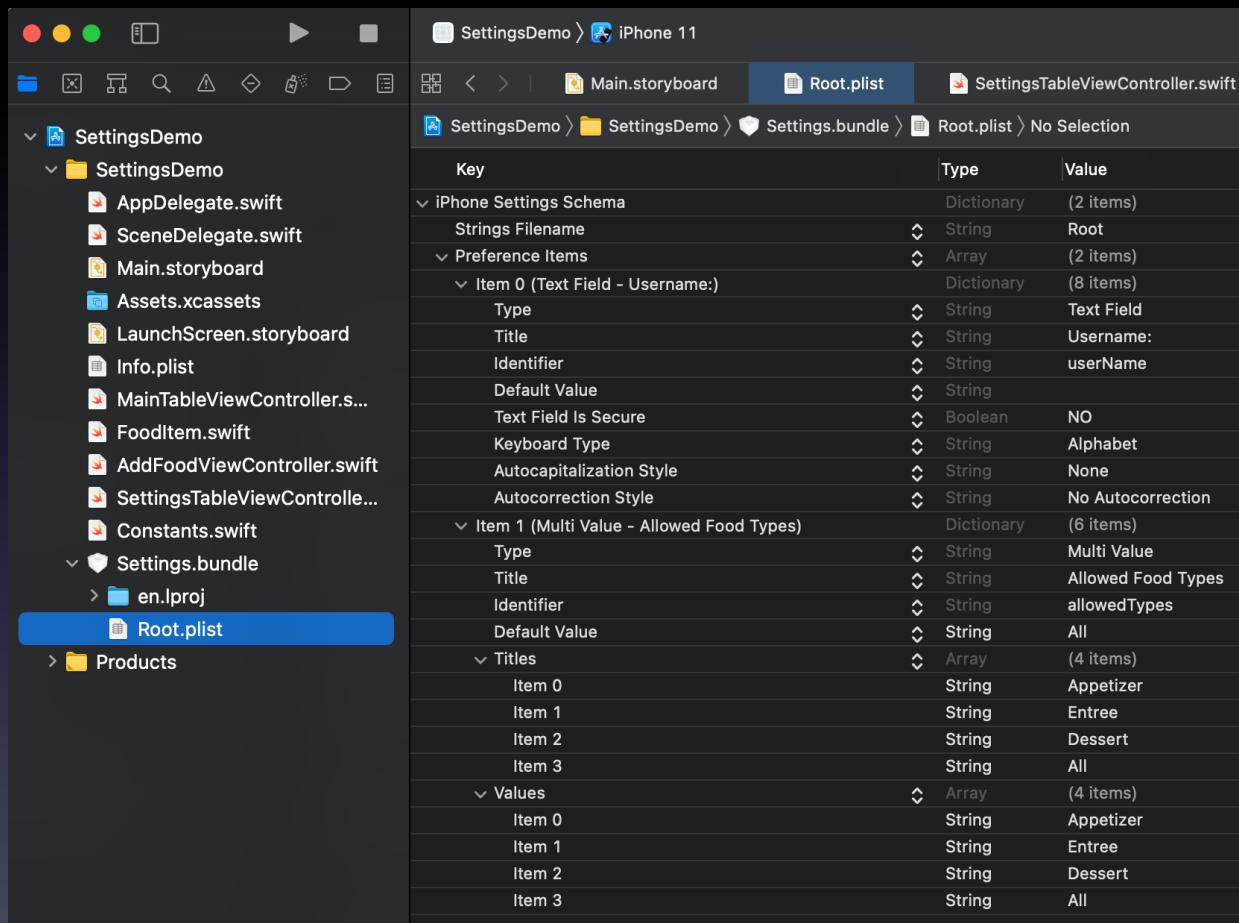


Settings Bundle

- Modify Root.plist to create settings

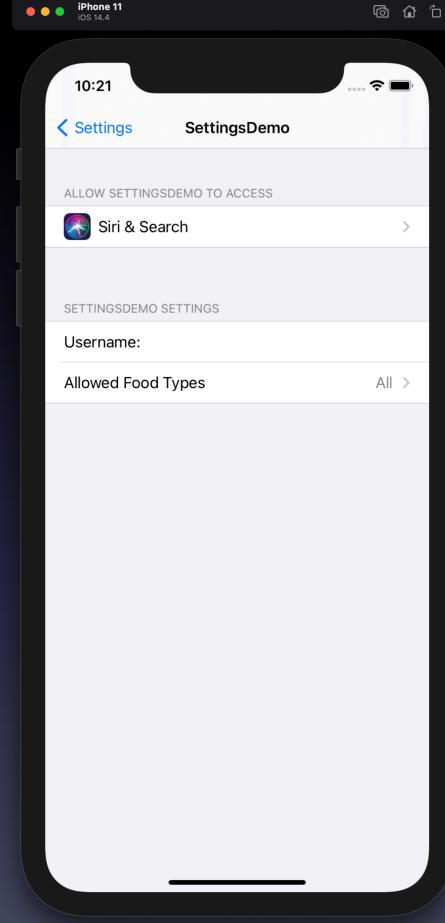


Settings Bundle



The screenshot shows the Xcode interface with the project "SettingsDemo" open. The left sidebar shows files like AppDelegate.swift, SceneDelegate.swift, Main.storyboard, and Root.plist. The main area displays the contents of Root.plist:

Key	Type	Value
iPhone Settings Schema	Dictionary	(2 items)
Strings Filename	String	Root
Preference Items	Array	(2 items)
Item 0 (Text Field - Username:)	Dictionary	(8 items)
Type	String	Text Field
Title	String	Username:
Identifier	String	userName
Default Value	String	
Text Field Is Secure	Boolean	NO
Keyboard Type	String	Alphabet
Autocapitalization Style	String	None
Autocorrection Style	String	No Autocorrection
Item 1 (Multi Value - Allowed Food Types)	Dictionary	(6 items)
Type	String	Multi Value
Title	String	Allowed Food Types
Identifier	String	allowedTypes
Default Value	String	All
Titles	Array	(4 items)
Item 0	String	Appetizer
Item 1	String	Entree
Item 2	String	Dessert
Item 3	String	All
Values	Array	(4 items)
Item 0	String	Appetizer
Item 1	String	Entree
Item 2	String	Dessert
Item 3	String	All



The right side shows a screenshot of an iPhone 11 running iOS 14.4. The Settings screen for the "SettingsDemo" app is displayed. It includes sections for "ALLOW SETTINGSDEMO TO ACCESS" (with a "Siri & Search" entry), "SETTINGSDEMO SETTINGS" (with a "Username:" field containing "userName"), and a "Allowed Food Types" section with a "All" button.

Accessing Settings Bundle

- Set in device's Global Settings
- Access in **UserDefaults** as before



```
let userName = UserDefaults.standard.string(forKey: "userName")
let allowedTypes = UserDefaults.standard.string(forKey: "allowedTypes")

UserDefaults.standard.set("Larry", forKey: "userName")
UserDefaults.standard.set("All", forKey: "allowedTypes")
```

- Warning: Device settings undefined until changed

Accessing Settings Bundle

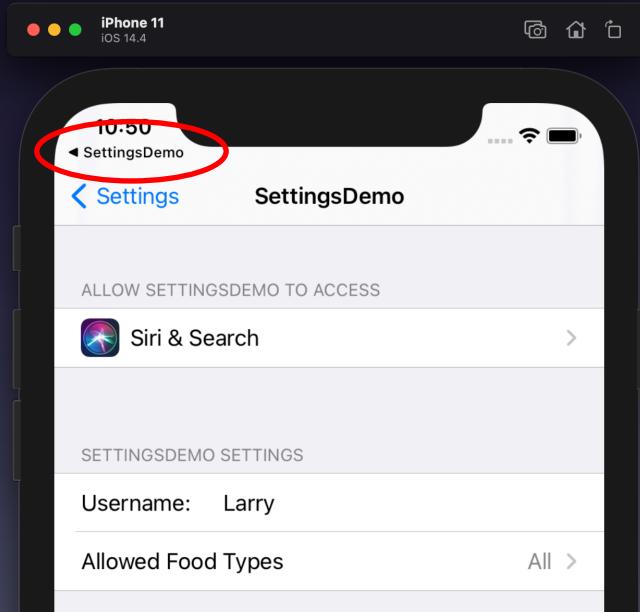
- How to detect when device settings changed?
- In SceneDelegate:

```
func sceneWillEnterForeground(_ scene: UIScene) {  
    let navController = self.window?.rootViewController as!  
        UINavigationController  
    if let settingsTVC = navController.topViewController as?  
        SettingsTableViewController {  
        settingsTVC.updateDeviceSettings() // Our method to update view  
    }  
}
```

- Or, send notification to app when device settings changed (next time...)

Open Device Settings from App

```
if let settingsURL = URL(string: UIApplication.openSettingsURLString) {  
    if UIApplication.shared.canOpenURL(settingsURL) {  
        UIApplication.shared.open(settingsURL, options: [:],  
            completionHandler: nil)  
    }  
}
```



Resources

- UserDefaults
 - developer.apple.com/documentation/foundation/user_defaults
- Settings Bundle
 - developer.apple.com/library/archive/documentation/Cocoa/Conceptual/UserDefaults/Preferences/Preferences.html