Data Input

Dr. Sarah Abraham

University of Texas at Austin
CS329e
Spring 2020

Model Layer of MVC

- Contains the data to be displayed
- Data can be:
  - Stored on device
  - Pulled down from a server
- Data displayed in app should be:
  - Personalized
  - Secure

User Defaults

- Storage on the device itself
- Can hold persistent key / value pairs
- Used for application configuration data
- Accessed via the NSUserDefaults singleton object:
  - `NSUserDefaults.standardUserDefaults()`
- Write back to hard drive using `synchronize()`

User Defaults Example

```
// Define keys for the values to store
let kUserIdKey = "userId"
let kTotalKey = "total"
let kNameKey = "name"

// Define the values to store
let userId = 900
let total = 1275.55
let name = "University of Texas"

// Get a reference to the global user defaults object
let defaults = NSUserDefaults.standardUserDefaults()

// Store various values
defaults.setInteger(userId, forKey: kUserIdKey)
defaults.setDouble(total, forKey: kTotalKey)
defaults.setObject(name, forKey: kNameKey)

defaults.synchronize()  // force the write to the device

// Retrieve the previously stored values
let retrievedUserId = defaults.integerForKey(kUserIdKey)
let retrievedTotal = defaults.doubleForKey(kTotalKey)
let retrievedName = defaults.objectForKey(kNameKey)
```

Read and Write Functions

- NS prefix signifies Foundation data type
- Can store:
  - NSData
  - NSString/String
  - NSNumber (UInt, Int, Float, Double)
  - NSDate
  - NSArray/Array, NSDictionary/Dictionary
- Write convenience functions:
  - `setBool(value: Bool, forKey defaultName: String)`
  - `setFloat`, `setInteger`, `setObject`, `setDouble`, `setURL`
- Read convenience functions:
  - `arrayForKey(defaultName: String)`
  - `boolForKey`, `dataForKey`, `dictionaryForKey`, `floatForKey`, `integerForKey`, `objectForKey`, `stringArrayForKey`, `stringForKey`, `doubleForKey`, `URLForKey`

Using User Default Methods

- Methods that return Bool, Int, Float and Double do not return Optionals
  - Return values appropriate to their type
- Methods that return `Any` must be cast to correct type before use
- API Reference [https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSUserDefaults_Class/]

What to Store in User Defaults?

- Relatively small amounts of data
- Related to the device user
- Settings / preferences etc
- **Not for sensitive data!**
  - User Defaults is in an unencrypted plist
  - Use **Keychain** to encrypt passwords, identifying information, etc

Plists

- Property lists organize data into named values and lists of values
- Converts to and from standard XML
- Convenient way to import standard app data to all devices
- Can be modified at runtime

Plist Values

- Plists can store:
  - Arrays/Dictionaries
  - Strings
  - NSData
  - Ints/Floats/Doubles
  - Booleans

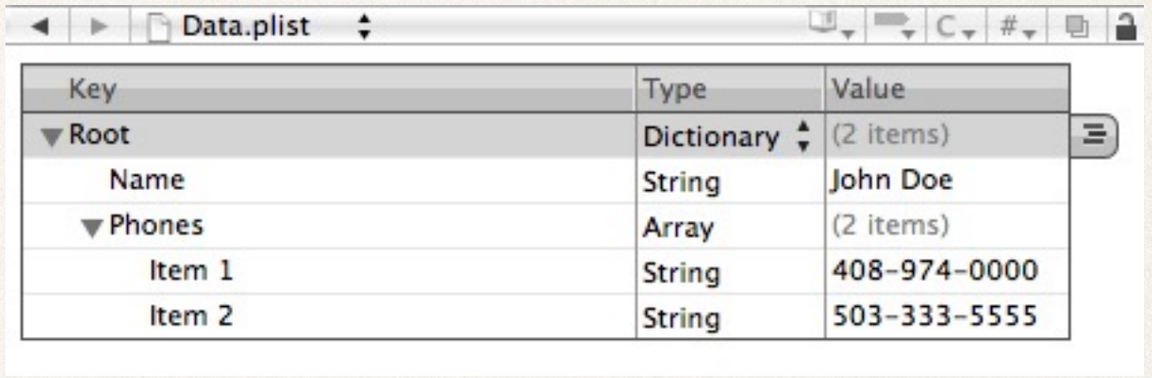
Plist Example

```
<?xml version="1.0" encoding=“UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">

<plist version="1.0">

<dict>
    <key>Name</key>
    <string>John Doe</string>
    <key>Phones</key>
    <array>
        <string>408-974-0000</string>
        <string>503-333-5555</string>
    </array>
</dict>
</plist>
```



Accessing Plists

1. Create reference from main bundle (e.g. the app's file structure)

2. Access as either NSArray or NSDictionary (depending on the plist)

```
let inputFile = Bundle.main.path(forResource:“plist_name",
ofType: “plist")

let inputArray = NSArray(contentsOfFile: inputFile!)

for item in inputArray as! [item_type] {

   /* Place plist items into appropriate object or data
   structures */

}
```

PlistDemo

What to Store in Plists?

- Data that is consistent across all users
- Data that is primarily read-only
- Information and variables that are used globally
- Settings that are related to the app rather than the user
  - e.g. level settings in a game or style settings for an app layout

Quiz Question!

- How do you access a plist at runtime?

A. NSArray(contentsOfFile:)

B. Bundle.main.path(forResource: ofType:)

C. NSUserDefaults.standardUserDefaults()