# Persistence in iOS

# Persistence

- Persistence is a way to preserve data so that it outlives application shutdown.

- Common way: files on local storage

# Means of Persistence

- Methods on Foundation classes
- NSUserDefaults
- NSCoding, NSKeyedArchiver/Unarchiver
- NSFileHandle
- NSFileManager
- Core Data

# Folders of the application

- Documents - NSDocumentDirectory
- Library/Application Support - NSApplicationSupportDirectory
- Library/Caches - NSCachesDirectory

- [NSFileManager URLsForDirectory: inDomains:].firstObject
- NSUserDomainMask
- NSHomeDirectory()
- NSTemporaryDirectory()

# NSURL

- Use NSURL to build paths to files/folders
- Try not to use NSString for that (unsafe)
- NSURL fileURLWithPath: isDirectory:
- lastPathComponent, path, URLByAppendingPathComponent: isDirectory:, URLByDeletingLastPathComponent

# Application folders sample

# Methods on Foundation classes

- NSString, NSData, NSArray, NSDictionary: <class>WithContentsOfFile: or <class>WithContentsOfURL: (e.g. stringWithContentsOfURL:).
- These types are representable in PList format.
- Return nil if unable to read file.
- UIImage imageNamed:, imageWithContentsOfFile:

# NSFoundation samples

# NSCoding

- Stores an arbitrary type in a file.
- - initWithCoder:(NSCoder *)
- - encodeWithCoder:(NSCoder *)
- Works mostly as a key-value container to read/write values of properties to store.

# NSKeyedArchiver/ NSKeyedUnarchiver

- + archiveRootObject: toFile:

- + unarchiveObjectWithFile:

# NSCoding sample

# NSUserDefaults

- Special type of plist file that stores ‘user settings’ according to name, but arbitrary plist-compatible data actually
- DO NOT store sensitive data there like tokens or passwords
- + standardUserDefaults
- - registerDefaults:
- <object>ForKey:
- set<Object>: forKey:

# NSUserDefaults

# sample

# NSFileManager

- -createDirectoryAtURL: withIntermediateDirectories: attributes: error:
- - createFileAtPath: contents: attributes:
- - removeItemAtURL: error:
- - copyItemAtURL: toURL: error:
- - moveItemAtURL: toURL: error:
- - fileExistsAtPath: isDirectory: - try not to check but to load and fail
- - contentsAtPath:

# NSFileHandle

- Wrapper over file descriptor: direct free access to bytes of underlying file. Able to read/write from/to sockets but on threads with Run loops.

- + fileHandleForReading[Writing|Updating]From[To]URL: error:

- seekToFileOffset:

- writeData:

- Reads or writes in background but must be started on a thread with Run Loop

# NSFileHandle sample

# Core Data

- Object Graph management system
- Tracks relationships between objects, monitors changes in objects.
- Able to fetch or store subgraphs of objects in SQLite, Memory cache, XML (OSX), your storage.
- Guarantees uniqueness and validates values.
- Supports migrations.

# Overview

- Model (NSManagedObjectModel)
- Persistent Store Coordinator (PSC) and Stores
- Contexts
- NSManagedObject
- NSFetchRequest
- NSPredicate

# How you work with it

- Create model (scheme) either is Xcode designer or manually in code
- Instantiate model from scheme file(s).
- Instantiate PSC with the model (scheme) and set a store on it. Assign model to PSC.Migrate if necessary.
- Create one or more Contexts
- Fetch objects
- Insert new objects, modify or delete existing ones.
- Save
- Merge changes to other contexts

# Core Data sample