Persistence & Core Data

How to Save Data

Intro

- Up until now, when we quit our app the data disappears.
- But most apps need to save data.
- Saving data is called “persistence”.

Wikipedia definition: "state that outlives the process that created it".

Download & Install Important Tools

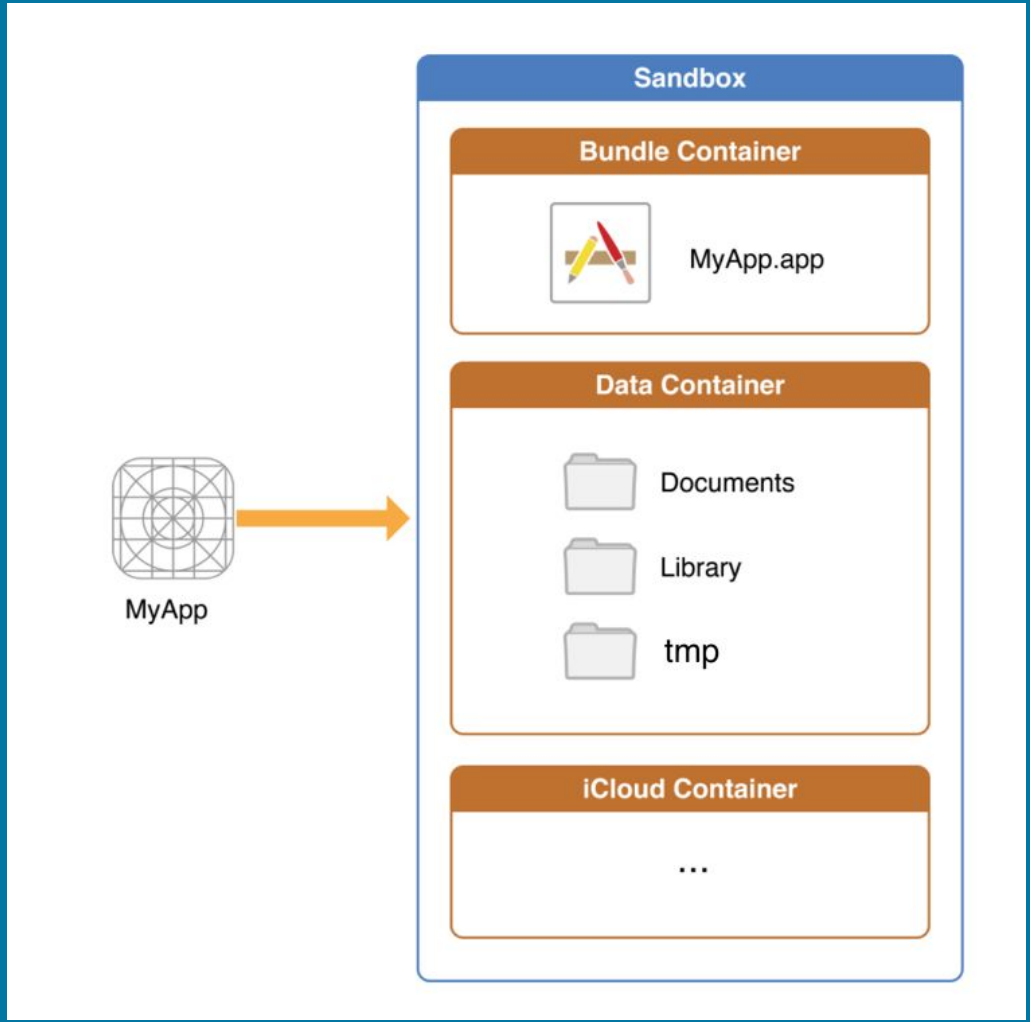
- OpenSim for opening files on the simulator http://bit.ly/2qdfKlb (Note: Add OpenSim to Login items)
- You need a SQLite viewer. Here's a free one http://sqlitebrowser.org

Data Persistence Options on iOS

- Writing to files
- NSUserDefaults (plist list file)
- Custom plist files
- Archiving (NSCoder/NSKeyedArchiver)
- Keychain (For secure storage)
- Direct SQL (SQLite, FMDB, etc.)
- Core Data

iOS Standard Directories

- iOS apps have very limited access to the file system.
- All iOS apps sit inside a sandbox directory.
- This sandbox directory has 2 or 3 (iCloud) container directories
- The Bundle Container stores app resources & is read only (changing is breaks code signing)
- Data Container which holds app & user data
- Data Container Directory is subdivided into Documents, Library, and Temp.
- Your app can request iCloud Container access at run time.


Sandbox
Bundle Container
MyApp.app
Data Container
Documents
Library
tmp
iCloud Container
...
MyApp

/Documents

- Store user generated content
- Exposed to users
- Backed up by iTunes & iCloud

/Documents/Inbox

- For files your app is asked to open. E.g. email attachments associated with your app.
- Your app can read/delete but not write.
- Backed up by iTunes/iCloud

Library/

- Top level dir for files that are *not* user data
- Standard subdirs *Preferences*, and *Caches*
- Can create custom subdirs.
- Files put here are not exposed to the user.
- Caches not backed up.

tmp/

- for writing and reading temp files
- might not be persisted between app launches
- not backed up

Steps for writing to disk

1. Get the location of a writable directory.
2. Append your target file’s name.
3. [optional] Check if file already exists.
4. Write to disk.

Writing to Files

- Many foundation classes have convenience methods for writing/reading to disk.
- Get path to file using NSFileManager
- Convert images to NSData to save.
- Write NSData to the file system.

FileSystemDemo

NSUserDefaults

- Intended to store simple user preferences
- Very similar to a dictionary, but it’s automatically persisted to disk
- Very simple to have synced via iCloud
- Does not easily store custom objects, but can be coerced to if you really want it
- Doesn’t perform well if larger data sets are added

NSUserDefaults

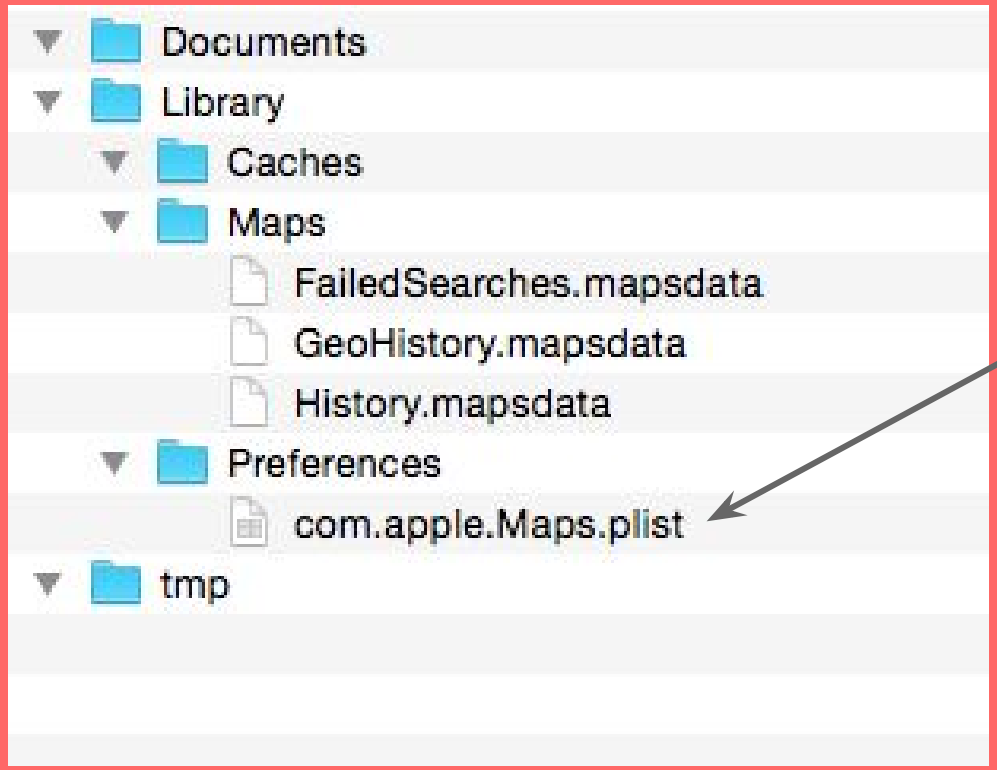
- User defaults can store **`NSData`**, **`NSString`**, **`NSNumber`**, **`NSDate`**, **`NSArray`**, and **`NSDictionary`** objects.

- Using **`setObject:forKey:`** and **`objectForKey:`**

- There are convenience methods for storing primitive types (they're boxed).

LIGHTHOUSE LABS

NSUserDefaults



NSUserDefaults

LIGHTHOUSE LABS

DEMO - NSUserDefaults

Databases

Author

id	name	age

Book

LIGHTHOUSE LABS

Databases

Author

id	name	age
1	John	44
2	Mary	34

Book

id	author_id	title

Databases

Author

id	name	age
1	John	44
2	Mary	34

Book

id	author_id	title
1	1	Learn iOS in 4 hrs
2	1	iOS for dummies

Core Data

- Core Data is an “Object Graph” and object lifecycle management framework. It *can* persist to a database.

- In other words, Core Data can be used to create your model layer independently of whether and how you want to persist it.

Core Data keeps your model layer consistent means it...

- Manages the relationships between your objects.
- Easy data validation when you want to save, update, or delete.
- Handles object deletion (and deletion of any associated objects).
- Has change-tracking and undo support.
- Has sophisticated query mechanism.
- Supports multi threading.

Core Data Things

- Managed object model
  - a collection of entity descriptions (schema?)
- Persistent Store Coordinator
  - maps between the storage (sqlite) and the MOM.
- Managed Object Context
  - a scratchpad to load objects into, modify, save.
- Managed Object
  - Just like any model objects except they are managed by core data.

NSManagedObjectContext

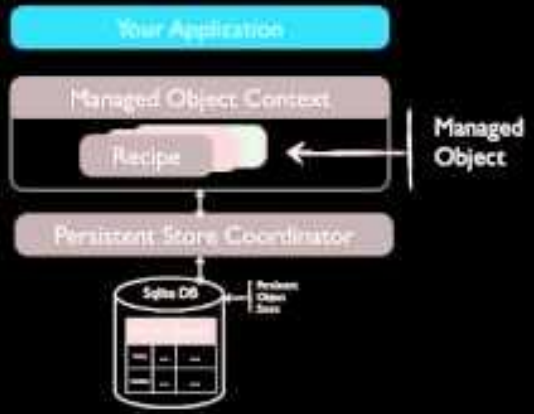
- Core Data's Scratch Pad.
- Handles CRUD (Create, Read, Update, Delete) operations.
- Handles Concurrency.
- Handles undo/redo.

NSManagedObject

- 99% of the time you'll use subclasses, rather than bare NSManagedObjects.
- State Information (was inserted, updated, deleted, etc).
- Lifecycle Hooks: methods called before/after important events (insertion, fetch, save).
- Validation.

CoreData

Accessing the Persistent Store



The Secret to Learning Core Data

- Use Apple's Core Data Snippets!
- Search for "Core Data Snippets" or go to https://developer.apple.com/library/content/documentation/DataManagement/Conceptual/CoreDataSnippets/Introduction/Introduction.html

Core Data CRUD Demo