

Chapter II

Introduction to iOS Application



I.Introduction to iOS Application

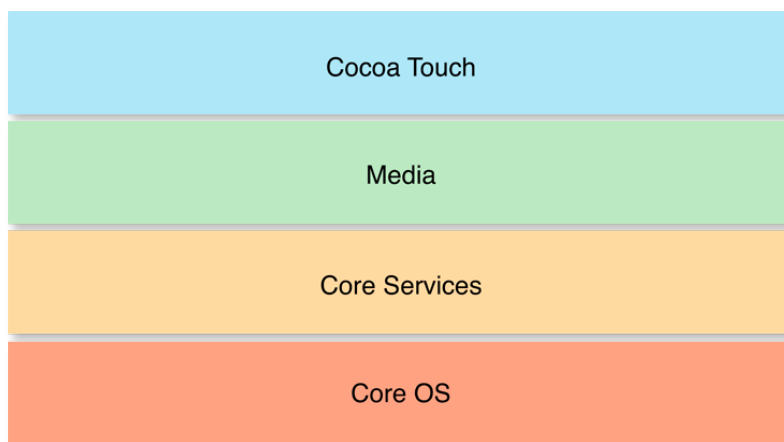
1. Human Interface Guide Line

- Refer to **Human Interface Guidelines**
- Keynote (Summary)

2. iOS App Architecture

At the highest level, iOS acts as an intermediary between the underlying hardware and the apps you create. Apps do not talk to the underlying hardware directly. Instead, they communicate with the hardware through a set of well-defined system interfaces.

The implementation of iOS technologies can be viewed as a set of layers, which are shown in Figure I-1. Lower layers contain fundamental services and technologies. Higher-level layers build upon the lower layers and provide more sophisticated services and technologies.



(Fig 2-1)

Note:

Cocoa Touch

Contains key frameworks defining appearance
Provide for touch-based input, multitasking, push notifications,
Documents handling, Storyboards and Auto layout etc.

Media

Graphics, audio, and video technologies you use to implement
multimedia experiences in your apps
UIKit , Core Graphics, Audio, Video , Animation, OpenGL ES,OpenAL ,Metal,
Media Player FW, AVFoundation, AirPlay

Core Services

Contains fundamental system services for app. Core Foundation Frameworks: Location, iCloud, Social Media & Networking

Core OS

contains the low-level features that most other technologies are built upon. Accelerate Framework: digital signal processing (DSP), linear algebra, and image-processing calculations ,

Bluetooth, External Accessories, Security Service

Kernel, low level, UNIX interface: Stand I/O, math Computation

3. Framework

The iOS technologies are packaged as frameworks. Apple delivers most of its system interfaces in special packages called frameworks. A framework is a directory that contains a dynamic shared library and the resources (such as header files, images, and helper apps) needed to support that library. To use frameworks, you add them to your app project from Xcode.
eg.Foundation.framework, UIKit.framework, MapKit.framework

4. Store and Distribution Channel

The App Store is a digital distribution platform for mobile apps on iOS. The apps can be downloaded directly to iOS devices or onto a personal computer via iTunes and offers an update service, e.g. if one application is updated, it will notify to update to existing users and allow to install.

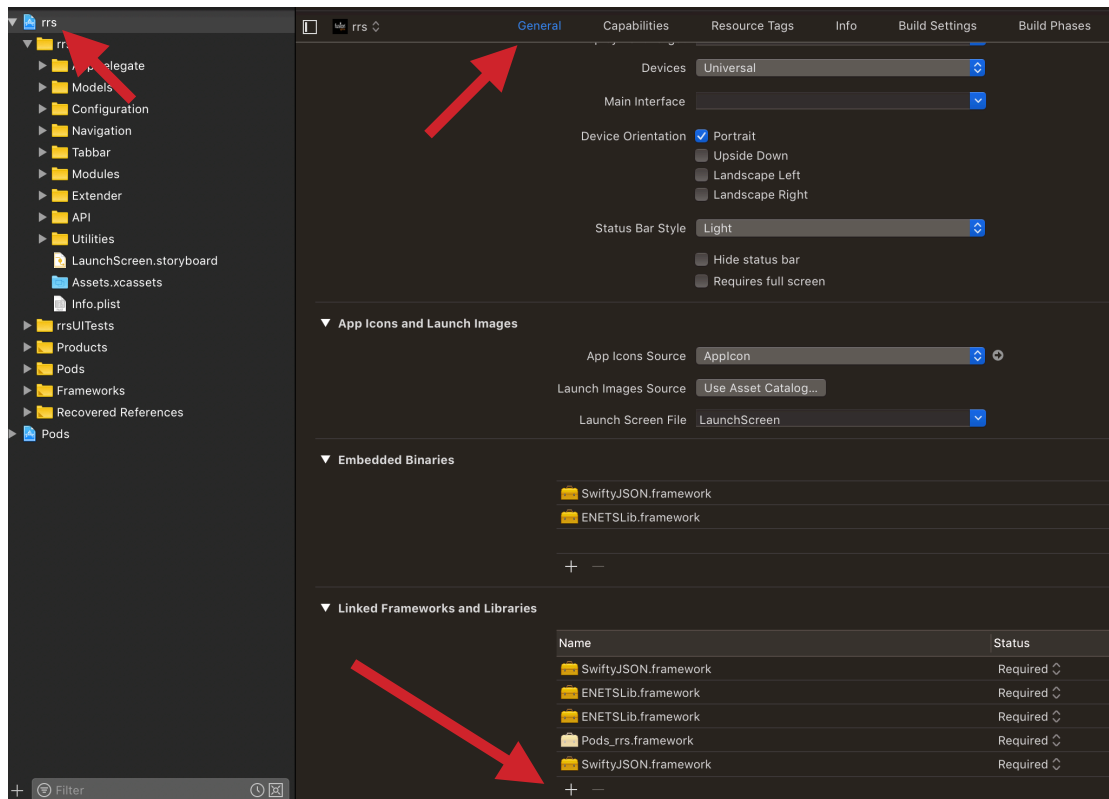
Enterprise Apps are in-house iOS apps developed for a certain organization and intended to use within organization

Register as iOS Developer:

<https://developer.apple.com/programs/enroll/>

Note: you can add new frameworks by selection

project -> General and then add new framework under Linked Frameworks and Libraries as shown below



Chapter II: Exercises

1. Create Free apple developer account and run your app on device

Important Note: You may need to enable you Mac for development, select iOS Device as destination and select your account in General Tab.

To setup with your account, you'll need to set up a provisioning profile to code sign your apps:

1. Open Xcode preferences (Xcode > Preferences...)
2. Click the 'Accounts' tab
3. Login with your Apple ID (+ > Add Apple ID...)

Once you've code signed your app, you should get a launch error fail. To get past this, all we have to do now is tell our iOS device to trust the certificate we code signed our app with. To do this, in your iOS device open the 'Settings' app and go to 'General > Device Management'. You'll see the email address associated with the Apple ID you used to code sign your app. Tap it, then tap 'Trust <your_email>':