

## Document Revision History

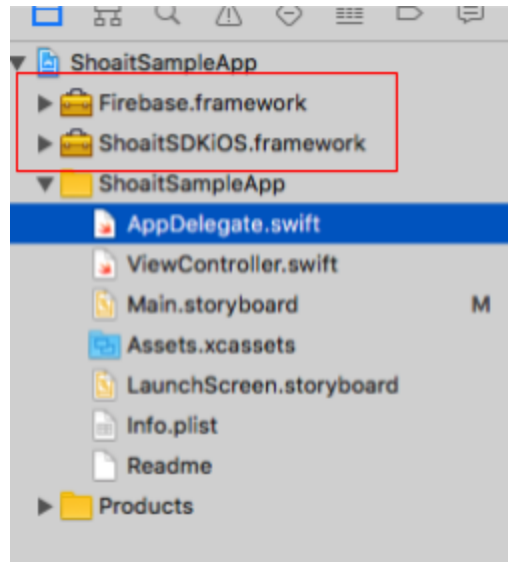
Revision	Date	Author	Description
0.1	2016-07-21	Aiden	初版
0.2	2016-08-30	Phil	修改1-3 depend 設定圖片

## iOS SDK Revision History

Revision	Date	Modified
1.0.0	2016-07-23	初版
1.1.0	2016-08-30	*加入loadHistory
1.1.1	2016-09-06	*loadHistory限制10筆
1.1.2	2016-09-07	*修正loadHistoy排序
1.2.0	2016-09-19	*加入loadChannels *使用Xcode8
1.3.0	2016-10-06	*加入 setSDKTimestamp 可讓user自行設定 離線訊息起始時間

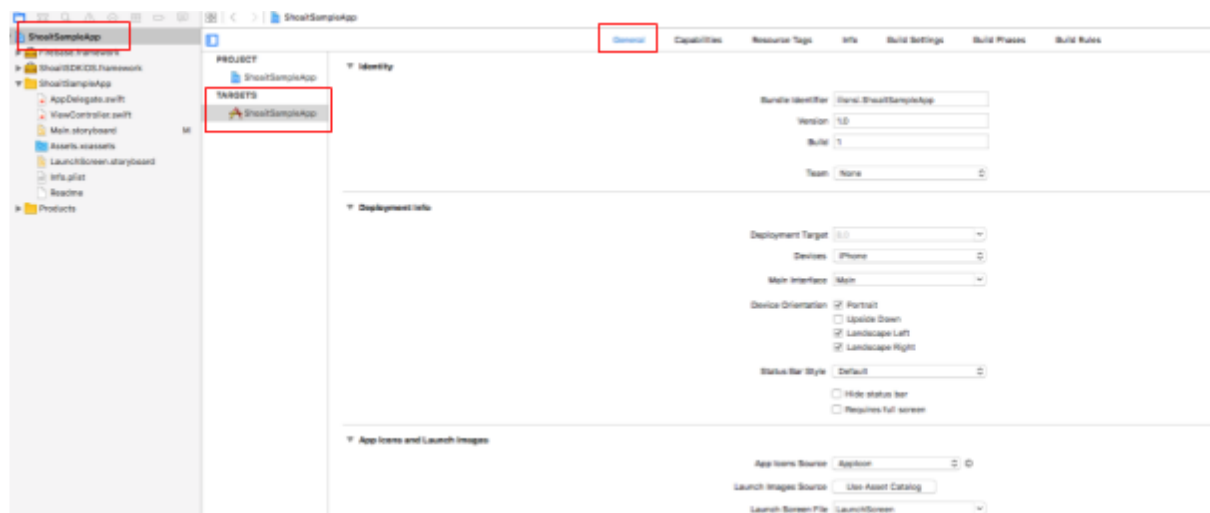
## 1. framework導入方法

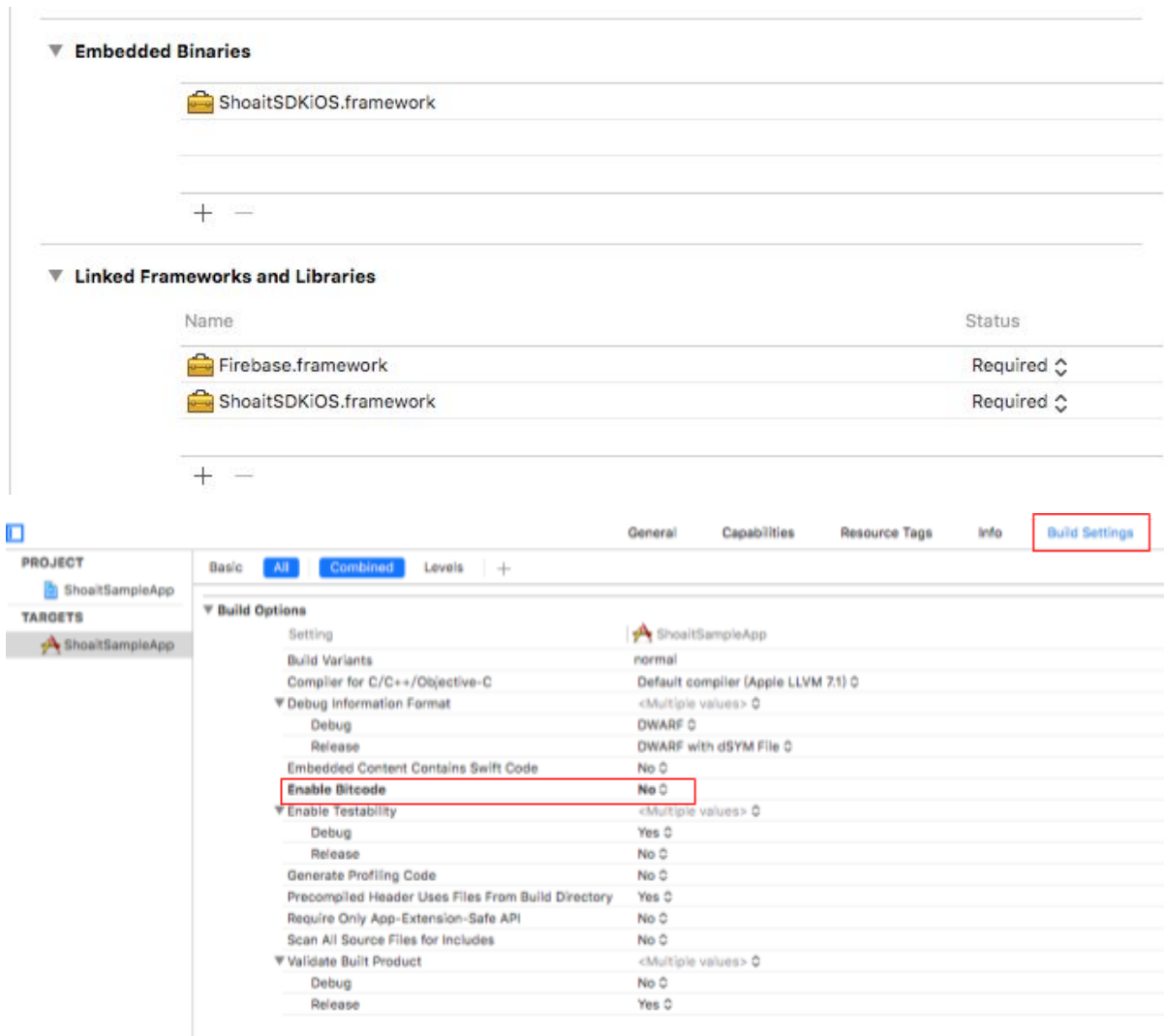
### 1-1.Xcode開新專案



### 1-2.拉入Firebase & shoaitSDKIOS framework

### 1-3.設定

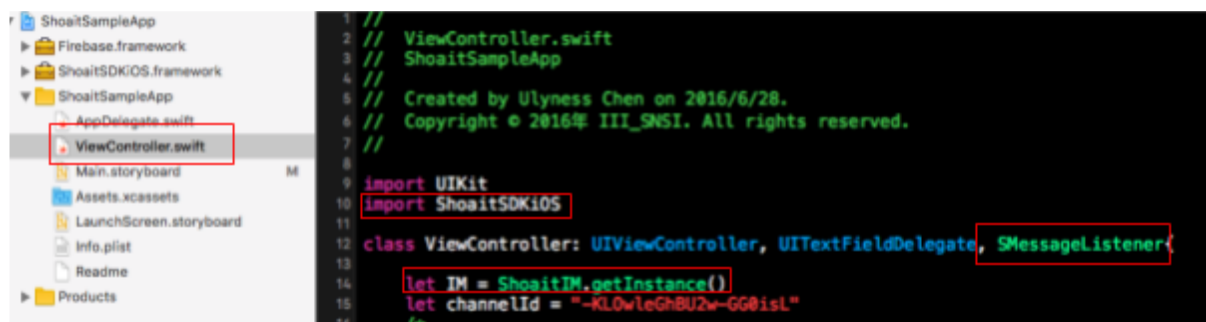




## 2. SDK基本運用

### 2-1. Open Controller page

import **ShoaitSDKiOS** & protocol **SMessageListener**



### 2-2. SDK init & add Listener

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a
    nib.

    IM.initIM("APP_ID")
    IM.addMessageLintener(self)

    // other ...
}

```

2-3. Add protocol func onMeassage() ->接收server chat訊息

```

func onMessage(message: SMessage) {
    // code...
}

```

2-4. Add protocol func onMeassageStatus() //處理收訊息狀態的地方

```

func onMessageStatus(status: SMessageStatus) {
    print("onMessageStatus = \(status)")
}

```

2-5. Login

```

IM.login(mail, password) { (err, auth) -> Void in
    print("err = \(err)")
    print("auth = \(auth?.uid)")
    //code...
}

```

2-6. Send message

```
IM.send(SMessage(channelId , type , message))
```

## 2-7 送訊息狀態

```
IM.setMessageStatus(status, message)
```

## 2-8 移除訊息監聽

```
override fun viewWillDisappear(animated: Bool) {  
    super.viewWillDisappear(animated)  
    IM.removeMessageLinsten(self)  
}
```

## 2-9 建立聊天室

```
func createChannel(type, uids, callback: (SError?) -> Void)
```

## 2-10 加入聊天室

```
func joinChannel(channelId: String, callback: (SError?) -> Void)
```

## 2-11 離開聊天室

```
func exitChannel(channelId: String, callback: (SError?) -> Void)
```

## 2-12 邀請進入聊天室

## 2-13 踢出聊天室

## 2-14 讀取歷史訊息與狀態

```
func loadHistory(channelId: String, timestamp: Int64 = -1, callback:  
([SMessage]) -> Void)
```

## 2-15 上傳檔案

```
self.IM.uploadFile(videoData, fileExtension: "mp4") { (err, path) -> Void in
    if err != nil {
        print("err = \(err)")
        return
    }
    print("path = \(path)")
    let message = SMessage(to: self.friendId, type: .VIDEO, message:
path!) //包裝指定的type
    self.IM.send(message)// 傳送
}

//fileExtension : 放副檔名
```

## 2-16 設定離線訊息時間

```
self.IM.setSDKTimestamp(NSDate().timeIntervalSince1970*1000
)
```

此API需在login之前調用