

Chapter 7

Understand iOS Delegate

Delegation & Callback

- ◎ Delegate เป็นหลักการพื้นฐานหนึ่งของ OOA&D
- ◎ การทำงานของ delegate ก็คือการที่ object A (ผู้เรียกใช้) assign งานๆ หนึ่งให้กับ object B ทำงาน (ผู้ถูกเรียก) โดยมองว่างานที่ object B ทำนั้นเป็นพุทธิกรรมของ A โดยที่ไม่มีการแก้ code ของ A แต่อาศัยการเรียก object B มาทำงานแทน
- ◎ หลักการ delegate นั้นมีให้เห็นทั่วไปในภาษาแบบ OOP ก็คือการเขียน code เพื่อเรียก method อีก method หนึ่งนั่นเอง
- ◎ โดยปกติแล้ว การ call method จะเป็นการทำงานแบบ synchronous คือรอให้ object B ทำงานเสร็จก่อน การทำงานจึง return กลับมาที่ object A แต่ในกรณีที่เป็นการทำงานแบบ multi-thread หรือ asynchronous แล้ว เราจะไม่รู้เลยว่าเมื่อไหร่ที่ object B จะทำงานเสร็จ เพราะฉะนั้นเราจึงต้องสร้างเทคนิคอื่นเพื่อรับรู้เพื่อให้รู้ว่า object B ทำงานเสร็จแล้ว
- ◎ หลักการการเรียกแบบ asynchronous และมีการ handle การ callback ใน Objective-C จะใช้คำว่า Delegate เลย

iOS Action Sheet & Alert

- ◎ ในภาษาอื่นนั้น แทนทุกภาษาที่รองรับ GUI จะสามารถแสดง dialog เพื่อให้ผู้ใช้สามารถโต้ตอบกับโปรแกรมได้เมื่อจำเป็นต้องเลือกหรือตัดสินใจ
- ◎ บน iOS นั้นจะมี dialog 2 แบบ คือ Action Sheet และ Alert ซึ่งทั้ง 2 แบบนั้นเป็น dialog ทั้งคู่แต่มีวัตถุประสงค์ไม่เหมือนกัน
 - ◎ Action Sheet นั้นจะเป็น dialog ที่เลื่อนขึ้นมาจากข้างล่างของหน้าจอ มักจะนำมาใช้เพื่อให้ user เลือกว่าจะทำ (action) อะไรต่อไป เช่น Create, Delete เป็นต้น
 - ◎ Alert นั้นเป็น dialog ที่มักจะใช้เพื่อแจ้งเตือน และ/หรือมี option ให้เลือกเพื่อไปเบิด โปรแกรมอื่น เช่น เมื่อ โปรแกรมต้องการใช้ internet แต่ iPhone เปิด Airplane mode โปรแกรมก็จะขึ้น Alert เตือนแล้วมีปุ่ม OK ให้กดเพื่อรับทราบ และมีปุ่ม Settings เพื่อให้ user มีทางเลือกที่จะไปเปิด Settings ขึ้นมาเพื่อปิด Airplane mode เป็นต้น
- ◎ Action Sheet และ Alert จะใช้เทคนิค delegate ในการ callback กลับมาที่ view controller ที่ใช้เรียกใช้ เพื่อบอกว่า user เลือกทางเลือกไหน และเราจะทำอะไรต่อกับทางเลือกนั้นเอง

Lab : Action Sheet & Alert (1/6)

◎ วัตถุประสงค์

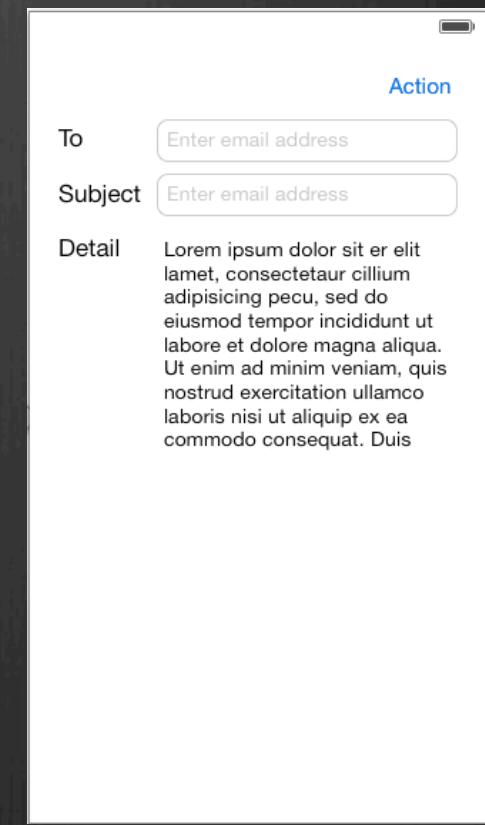
- ◎ เพื่อให้เข้าใจการใช้งาน delegate ที่ถูก implement ไว้แล้วใน iOS SDK
- ◎ เพื่อให้เข้าใจการใช้งาน UIAlertView และ UIActionSheet รวมทั้ง delegate ของ class ทั้ง 2 ตัว

◎ ขั้นตอน

- ◎ สร้าง project ใหม่
- ◎ เขียนโปรแกรมเพื่อแสดง Action Sheet และ Alert
- ◎ เขียนโปรแกรมเพื่อ handle เมื่อ user เลือก option บน Action Sheet และ Alert View

Task : Create Project (2/6)

1. จาก Xcode สร้าง project ใหม่โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “SendMail” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. เปิด Main.storyboard และวางแผน control ตามรูป
 - Label เปลี่ยน title เป็น “To”
 - Label เปลี่ยน title เป็น “Subject”
 - Label เปลี่ยน title เป็น “Detail”
 - วาง Text Field 2 ตัว
 - วาง Text View 1 ตัว
 - Button เปลี่ยน Title เป็น “Action”



Task : Implement Delegate (3/6)

5. เปลี่ยน editor mode เป็น Assistance Editor และลาก outlet และ action
 - Textbox แรกเป็น Outlet ตั้งชื่อว่า “txtTo”
 - Textbox ที่ 2 เป็น Outlet ตั้งชื่อว่า “txtSubject”
 - Button เป็น Action ตั้งชื่อว่า “btnActionTouched”
6. เปิดไฟล์ ViewController.h เพิ่ม code ตามนี้

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
    <UIActionSheetDelegate, UIAlertViewDelegate>

@property (weak, nonatomic) IBOutlet UITextField *txtTo;
@property (weak, nonatomic) IBOutlet UITextField *txtSubject;

- (IBAction)btnActionTouched:(id)sender;

@end
```

Task : Implement Delegate (4/6)

7. เขียน code ใน method “btnActionTouched”

```
- (IBAction)btnActionTouched:(id)sender
{
    if ([self.txtTo.text isEqualToString:@""]){
        [[[UIAlertView alloc] initWithTitle:@"Email Require"
                                     message:@"ยังไม่ได้ระบุ Email ปลายทาง"
                                     delegate:nil
                                     cancelButtonTitle:@"OK"
                                     otherButtonTitles:nil, nil] show];
    } else {
        [self.txtTo resignFirstResponder];
        NSString * sendTitle = [NSString stringWithFormat:@"Send to %@", self.txtTo.text];
        UIActionSheet * actionSheet =
            [[[UIActionSheet alloc] initWithTitle:@"Send Email"
                                         delegate:self
                                         cancelButtonTitle:@"Don't Send"
                                         destructiveButtonTitle:sendTitle
                                         otherButtonTitles:nil, nil];
        [actionSheet showInView:self.view];
    }
}
```

Task : Create Project (5/6)

8. เพิ่ม method “actionSheet:clickedButtonAtIndex:” ในไฟล์ ViewController.m

```
- (void)actionSheet:(UIActionSheet *)actionSheet
    clickedButtonAtIndex:(NSInteger)buttonIndex
{
    if (buttonIndex == 0) {
        [[[UIAlertView alloc] initWithTitle:@"Done"
                                     message:@"Do you want to save copy?"
                                     delegate:self
                                     cancelButtonTitle:@"No"
                                     otherButtonTitles:@"Yes", nil] show];
    }
}
```

Task : Create Project (6/6)

9. เพิ่ม method “alertView:clickedButtonAtIndex:” ในไฟล์ ViewController.m

```
- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex
{
    NSString * alertMsg;
    switch (buttonIndex) {
        case 0:
            alertMsg = @"Message discarded.";
            break;

        default:
            alertMsg = @"Message saved.";
            break;
    }
    [[[UIAlertView alloc] initWithTitle:@"Save..." message:alertMsg delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil, nil] show];
}
```

Alert View Style

เราสามารถเปลี่ยน style ของ UIAlertView เป็นแบบต่างๆ ได้ (ตั้งแต่ iOS 5) โดย set property alertViewStyle

เช่น alertView.alertViewStyle = UIAlertViewStyleLoginAndPasswordInput;

UIAlertViewStyle ที่เป็นไปได้ คือ

- ⊛ UIAlertViewStyleDefault
- ⊛ UIAlertViewStylePlainTextInput
- ⊛ UIAlertViewStyleSecureTextInput
- ⊛ UIAlertViewStyleLoginAndPasswordInput

และอ่านค่าที่ป้อนใน text box สามารถอ้างได้จาก

[alert textFieldAtIndex : (int)] ใน delegate method “alertView:clickedButtonAtIndex:”



Note: iOS 7 SDK ไม่รองรับการ add sub view ใน Alert view ออกต่อไป

Hint

Note1 : method “actionSheet:clickedButtonAtIndex:” เป็น method ที่ action sheet จะ callback กลับมาเมื่อ user เลือกที่ปุ่ม ใดๆ ของ action sheet โดยที่ action sheet จะ callback กลับมาที่ method นี้ก็ต่อเมื่อเราระบุ **delegate : self** เอาไว้ใน method “initWithTitle” ของ action sheet (ตามข้อ 7.)

Note2 : method “actionSheet:clickedButtonAtIndex:” นั้นค่อนข้างยาว โอกาสที่จะพิมพ์ผิดมีสูง และที่สำคัญภาษา Objective-C นั้นเราราสามารถเขียน method ในไฟล์ .m โดยที่ไม่ต้องประกาศในไฟล์ .h ก็ได้ เพราะจะนั้นถ้าพิมพ์ผิด (case-sensitive) โปรแกรมจะไม่ error แต่ delegate callback จะไม่ทำงานแทน

วิธีแก้ปัญหาแบบแรก คือ พิมพ์แค่ส่วนต้นของ method แล้วให้ Xcode เติม code ส่วนที่เหลือให้ โดยให้จำว่าถ้าเป็น delegate ของ Alert view ชื่อ method จะขึ้นต้นด้วย **-(void)alert...** ส่วน Action sheet จะขึ้นต้นด้วย **-(void)actionSheet...**

วิธีที่สอง คือ เมื่อเรารู้ว่า method นี้เป็น method ของ delegate UIActionSheetDelegate ก็ใช้วิธีเปิดดู code ของ delegate ดู โดย click ขวาที่ UIActionSheetDelegate และเลือก “Jump to Definition” แล้ว copy method ที่ต้องการมาเลย