

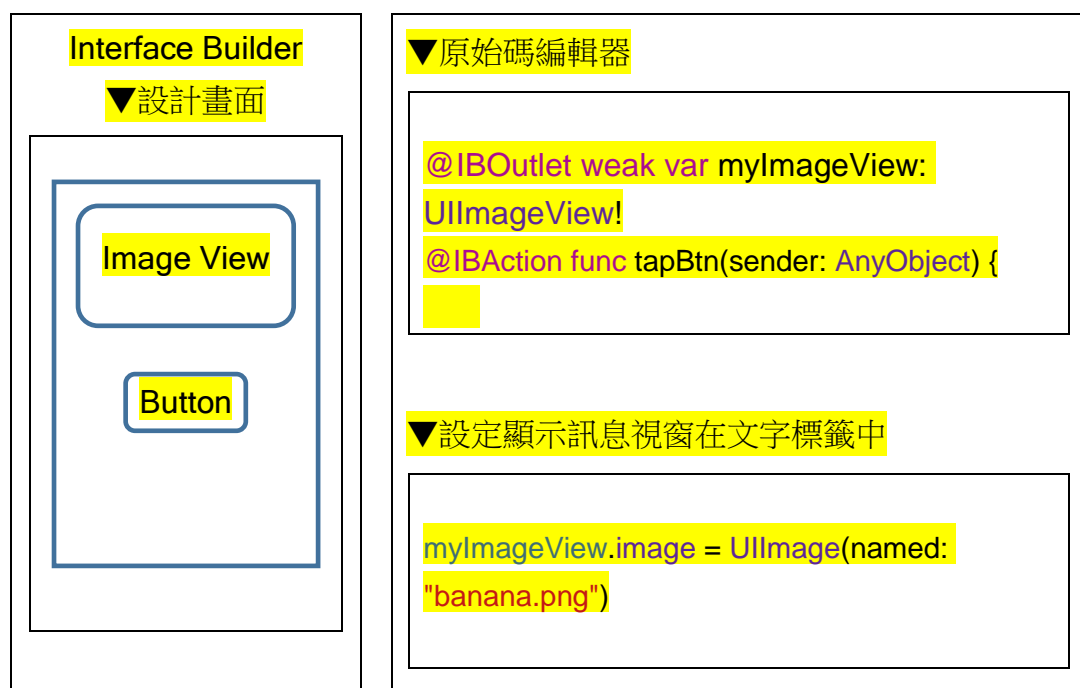
CHAPTER 10-1

Master-Detail Application + JSON :

使用 Master-Detail 顯示 JSON 資料

學習概念：

1. 首先用 IB 建立【圖像方塊】。
2. 將【操作按鈕】及【圖像方塊】與【程式碼】連結。
3. 最後在實作檔中相關程式，於處理載入後所觸發的事件，也就是撰寫利用【操作按鈕】結合【圖像方塊】，直接顯示在【設計畫面】上的程式。

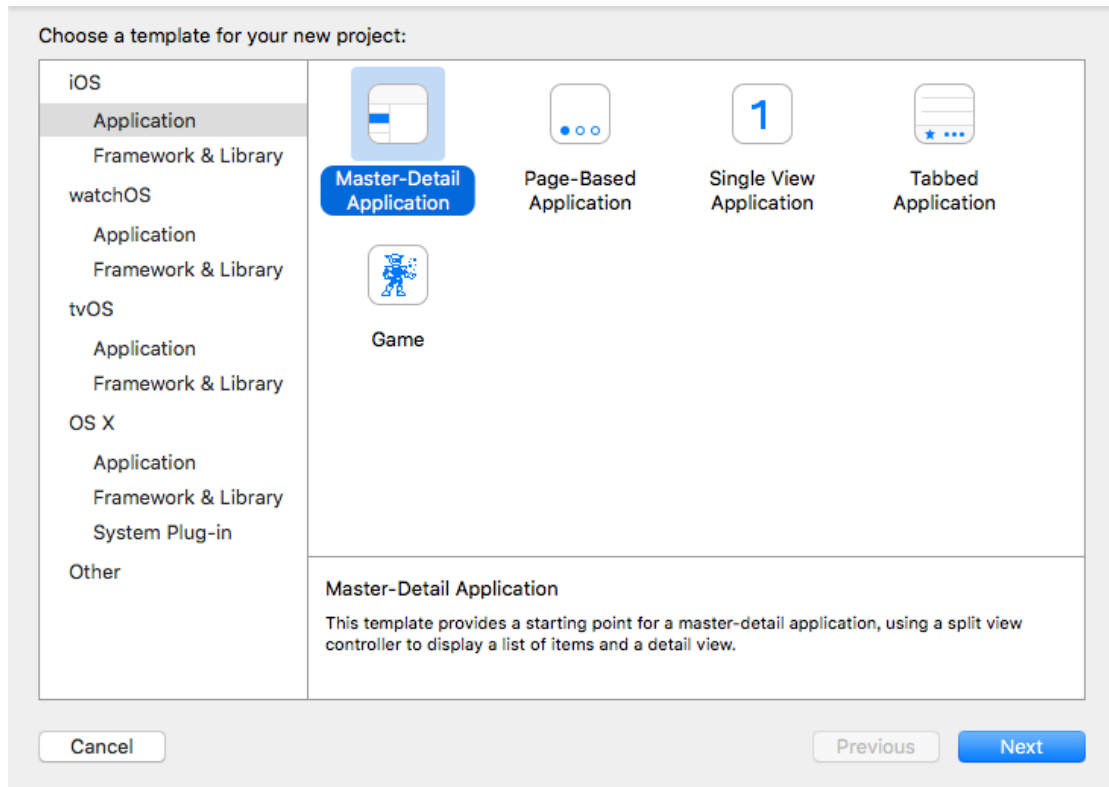


【執行結果】

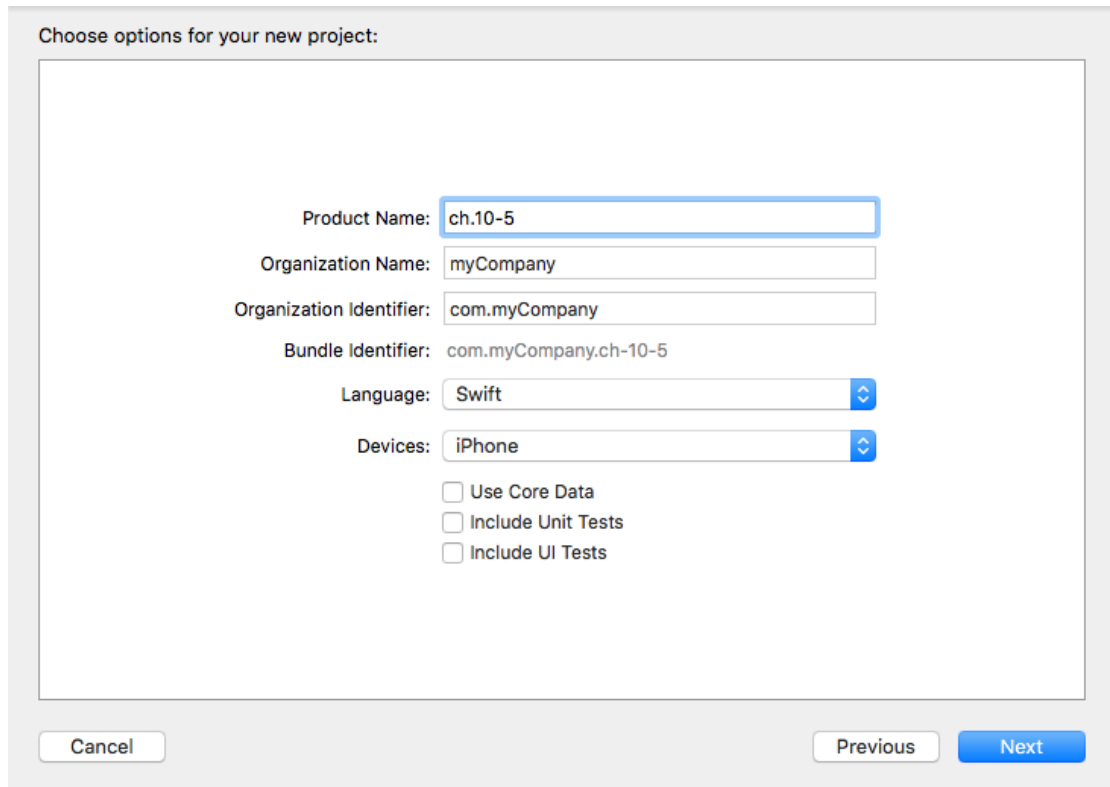
當 App 執行後，預先顯示 apple.png 圖片，點選【操作按鈕】後，讓【圖像方塊】顯示 banana.png 圖片在〈設計畫面〉中。

Step.1

開啟 **xcode** 時會出現的畫面，點選 **iOS** 下的【**Application**】，接著右視窗選擇【**Master-Detail Application**】，點選【**Next**】選項後進入設定的基本視窗。



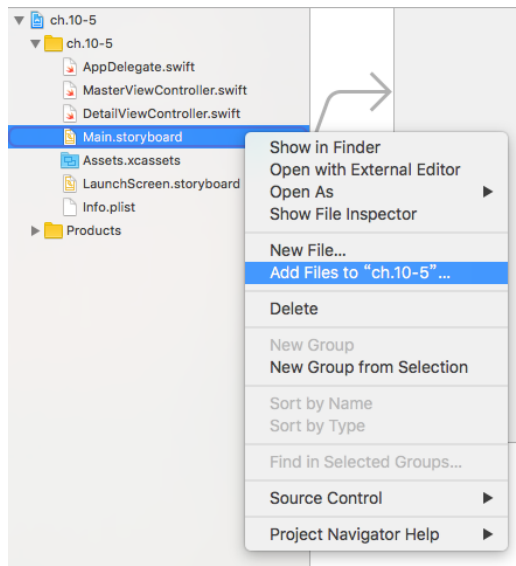
檔名及名稱設定，請將【**Product Name**】設定為 **ch.10-5**



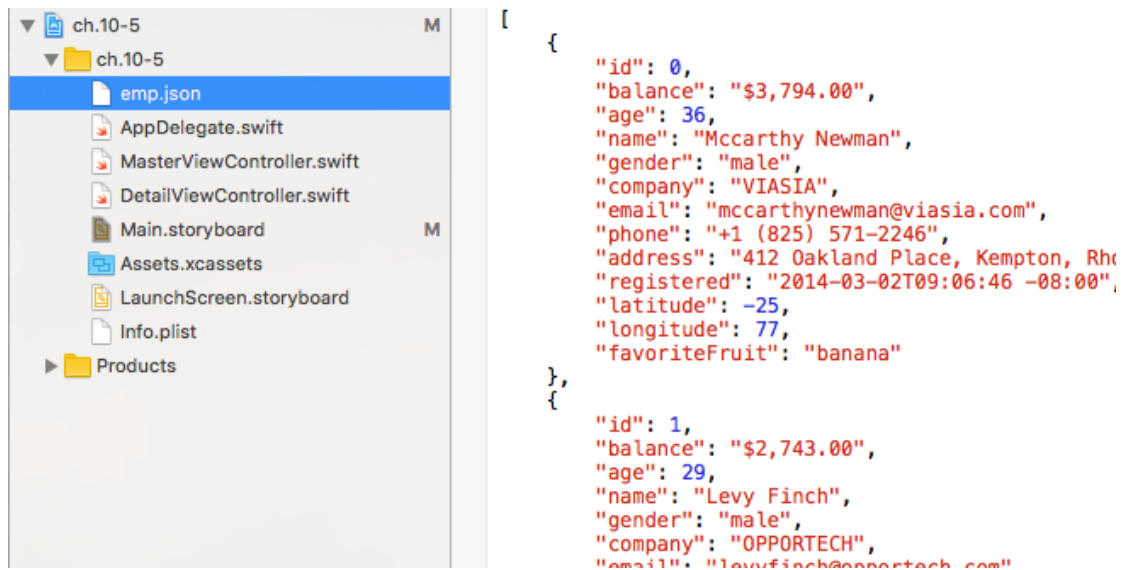
Step.2

附加檔案 json 於專案中

於左側專案瀏覽器中 點選右鍵 即可附加檔案 「Add Files to “專案名稱”」，選擇本練習所要使用的 emp.json 檔案，



載入後，可點選 emp.json 檔案，即可瀏覽檔案內的 json 內容。



Step.3.

修改範本資料，讓他顯示我們想要顯示的資料

開啟 MasterViewController.swift

(1) 於 **func viewDidLoad()** 增加讀檔與 json 語法解析之程式，解析之後的資料仍放置於此範本所使用的 **objects** 物件內。

```
let path = NSBundle.mainBundle().pathForResource("emp", ofType: "json")
let jsonData = NSData(contentsOfFile: path!)

do {
    self.objects = try NSJSONSerialization.JSONObjectWithData(jsonData!,
options: NSJSONReadingOptions.MutableContainers) as! [AnyObject]
} catch {
    print("Fetch failed: \(error as NSError).localizedDescription")
}
```

在做轉場效果，需要將主表格中的資料帶入後面明細頁，此時資料型態也需要由 **NSDate** 改為 **NSString**

```
let object = objects[indexPath.row] as! NSDate
改為 let object = objects[indexPath.row] as! NSDictionary
```

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if segue.identifier == "showDetail" {
        if let indexPath = self.tableView.indexPathForSelectedRow {
            //let object = objects[indexPath.row] as! NSDate

            let object = objects[indexPath.row] as! NSDictionary

            let controller = (segue.destinationViewController as!
 UINavigationController).topViewController as! DetailViewController
            controller.detailItem = object
        }
    }
}
```

```

        controller.navigationItem.leftBarButtonItem =
self.splitViewController?.displayModeButtonItem()
        controller.navigationItem.leftItemsSupplementBackButton = true
    }
}
}

```

- (2) 在表格中對每列資料給值時，修改原先的程式，將 NSDate 改為使用 NSDictionary，並使用 objectForKey("name") 取出該筆資料姓名的資料，並將其做型別轉換，轉為 String

```
let object = objects[indexPath.row] as! NSDate
```

改為 `let object = objects[indexPath.row] as! NSDictionary`

```
cell.textLabel!.text = object.description
```

改為 `cell.textLabel!.text = object.objectForKey("name") as! NSString as String`

```

override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath)
-> UITableViewCell {
    let cell = tableView.dequeueReusableCellWithIdentifier("Cell", forIndexPath: indexPath)

    //let object = objects[indexPath.row] as! NSDate
    //cell.textLabel!.text = object.description

    let object = objects[indexPath.row] as! NSDictionary
    cell.textLabel!.text = object.objectForKey("name") as! NSString as String

    return cell
}

```

```

class MasterViewController: UITableViewController {

    var detailViewController: DetailViewController? = nil
    var objects = [AnyObject]()

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

        let addButton = UIBarButtonItem(barButtonSystemItem: .Add, target: self, action: "insertNewObject:")
        self.navigationItem.rightBarButtonItem = addButton
        if let split = self.splitViewController {
            let controllers = split.viewControllers
            self.detailViewController = (controllers[controllers.count-1] as! UINavigationController).
                topViewController as? DetailViewController
        }

        let path = NSBundle.mainBundle().pathForResource("emp", ofType: "json")
        let jsonData = NSData(contentsOfFile: path!)

        do {
            self.objects = try NSJSONSerialization.JSONObjectWithData(jsonData!, options: NSJSONReadingOptions.
                MutableContainers ) as! [AnyObject]
        } catch {
            print("Fetch failed: \(error as NSError).localizedDescription)")
        }
    }
}

```

```

// MARK: - Segues

override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
    if segue.identifier == "showDetail" {
        if let indexPath = self.tableView.indexPathForSelectedRow {
            //let object = objects[indexPath.row] as! NSDate
            let object = objects[indexPath.row] as! NSDictionary
            let controller = (segue.destinationViewController as! UINavigationController).topViewController as!
                DetailViewController
            controller.detailItem = object
            controller.navigationItem.leftBarButtonItem = self.splitViewController?.displayModeButtonItem()
            controller.navigationItem.leftItemsSupplementBackButton = true
        }
    }
}

// MARK: - Table View

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {
    return 1
}

override func tableView(tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
    return objects.count
}

override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
    UITableViewCell {
    let cell = tableView.dequeueReusableCellWithIdentifier("Cell", forIndexPath: indexPath)

    //let object = objects[indexPath.row] as! NSDate
    //cell.textLabel!.text = object.description

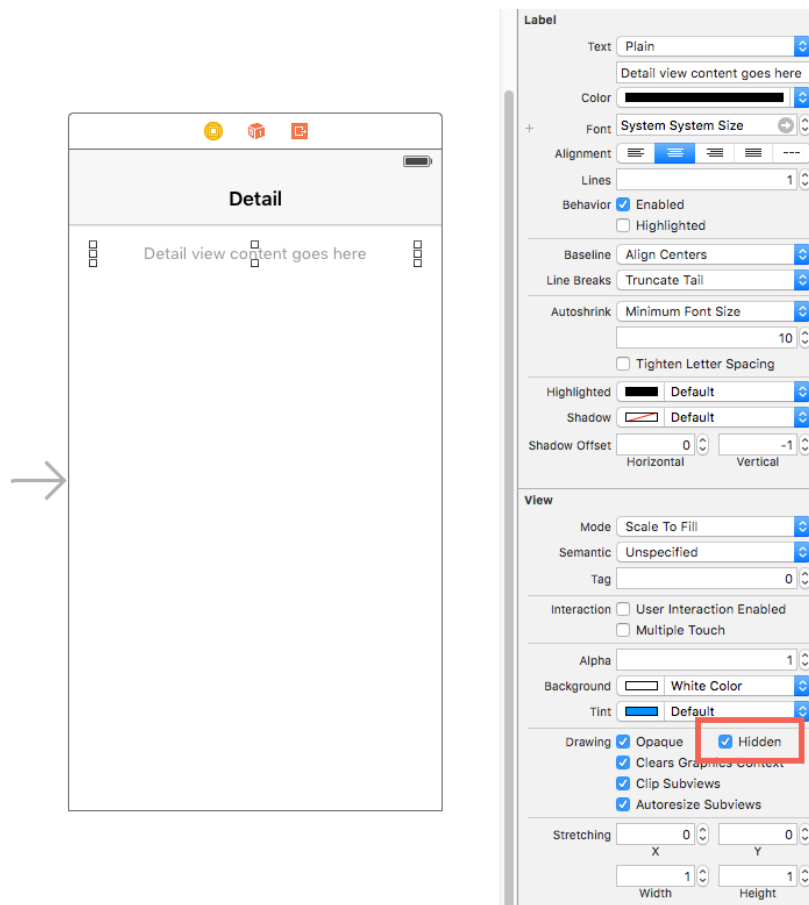
    let object = objects[indexPath.row] as! NSDictionary
    cell.textLabel!.text = object.objectForKey("name") as! NSString as String

    return cell
}

```

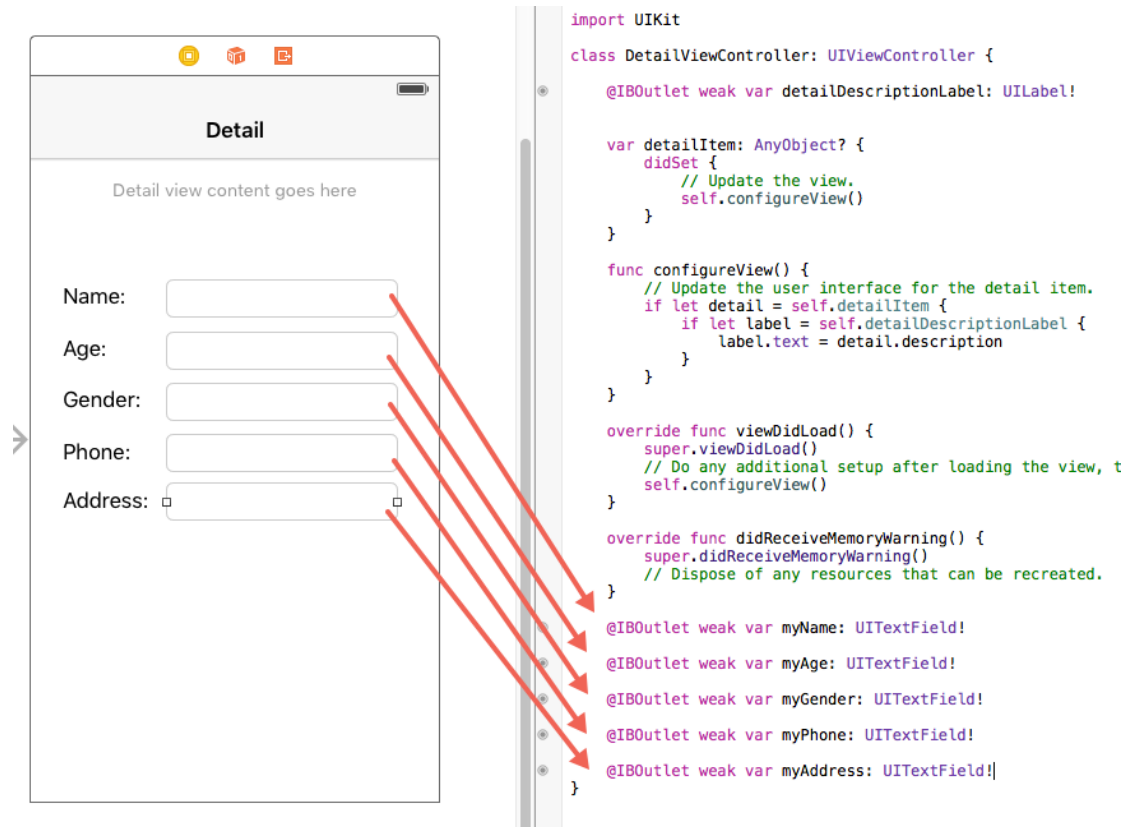
Step.4.

開啟 Main.storyboard，於 Detail 畫面中，建置好員工的個人資料檔呈現畫面。並可先將範本中原先的 Label “Detail view content goes here” 先透過屬性設定將其隱藏



Step.5.

於 Detail 畫面中建立五個 TextField，將新建立的五個文字欄位連結其名稱



Step.6.

於 DetailViewController.swift 中設定顯示明細資料

透過 `self.detailItem?.objectForKey("name")` 的方式取出 NSDictionary 中取出 key 為 "name" 的儲存值。

對於文字欄位必須進行型別轉換，轉換為 String

對於數字欄位則透過 stringValue 方式取出文字型數值

```
self.myName.text = self.detailItem?.objectForKey("name") as! NSString as String
let varAge = self.detailItem?.objectForKey("age")
self.myAge.text = varAge!.stringValue
self.myGender.text = self.detailItem?.objectForKey("gender") as! NSString as String
self.myPhone.text = self.detailItem?.objectForKey("phone") as! NSString as String
self.myAddress.text = self.detailItem?.objectForKey("address") as! NSString as
String
```



```

import UIKit

class DetailViewController: UIViewController {

    @IBOutlet weak var detailDescriptionLabel: UILabel!

    var detailItem: AnyObject? {
        didSet {
            // Update the view.
            self.configureView()
        }
    }

    func configureView() {
        // Update the user interface for the detail item.
        if let detail = self.detailItem {
            if let label = self.detailDescriptionLabel {
                label.text = detail.description
            }
        }
    }

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

        self.myName.text = self.detailItem?.objectForKey("name") as! NSString as String
        let varAge = self.detailItem?.objectForKey("age")
        self.myAge.text = varAge!.stringValue
        self.myGender.text = self.detailItem?.objectForKey("gender") as! NSString as String
        self.myPhone.text = self.detailItem?.objectForKey("phone") as! NSString as String
        self.myAddress.text = self.detailItem?.objectForKey("address") as! NSString as String
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var myName: UITextField!

    @IBOutlet weak var myAge: UITextField!

    @IBOutlet weak var myGender: UITextField!

    @IBOutlet weak var myPhone: UITextField!

    @IBOutlet weak var myAddress: UITextField!
}

```

Step.7

執行程式。

可以看到畫面中主表格顯示員工的姓名，點選後在明細頁會顯示其相關資料。

<<本練習中並未完成新增之功能，若點選新增之功能則會執行錯誤>>

