

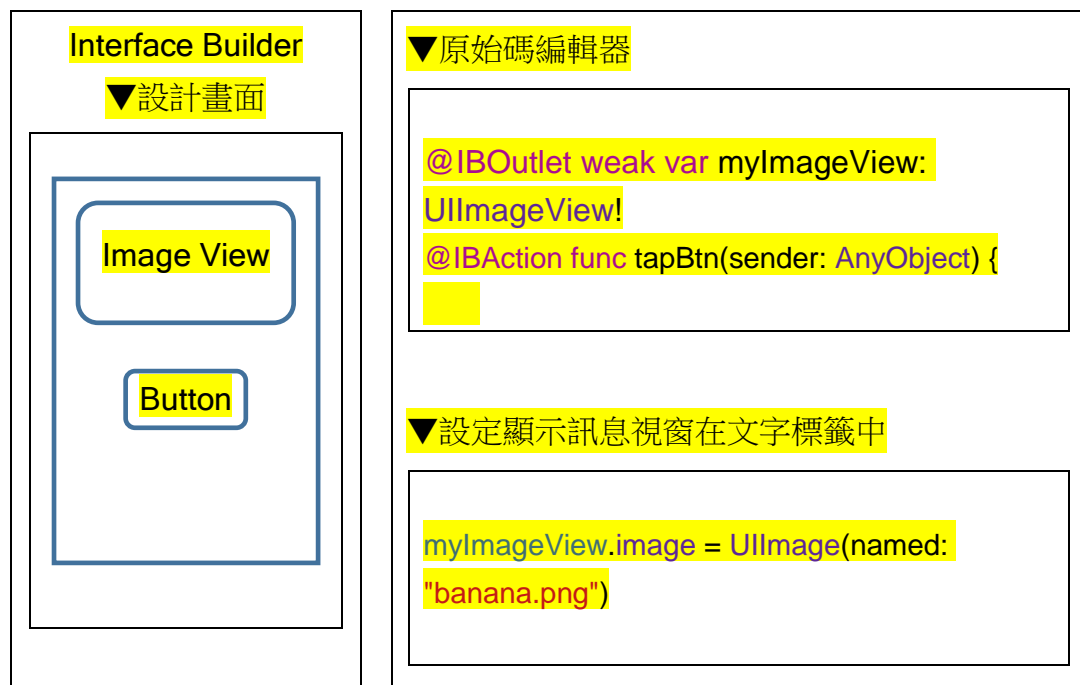
CHAPTER 10-1

Master-Detail Application + JSON on web :

使用 Master-Detail 顯示網路上的 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。

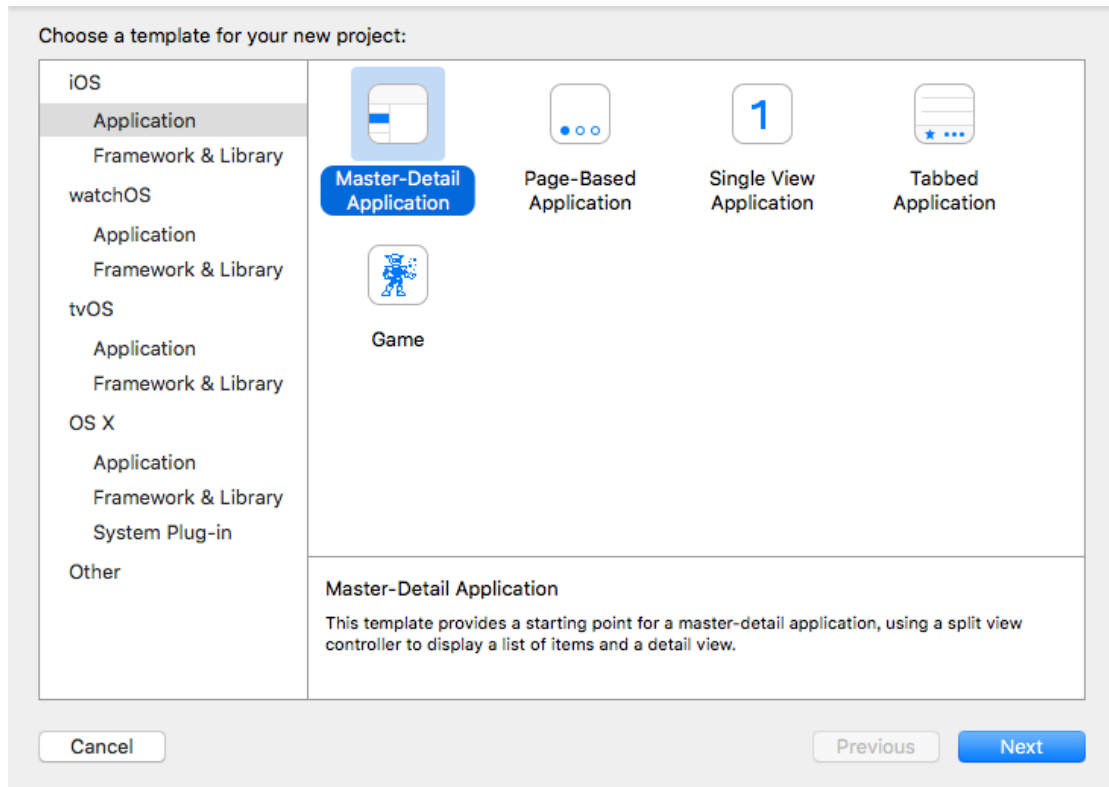


【執行結果】

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.10-6**

Choose options for your new project:

Product Name:

Organization Name:

Organization Identifier:

Bundle Identifier:

Language:

Devices:

☐ Use Core Data

☐ Include Unit Tests

☐ Include UI Tests

Step.2

政府資料開放平臺 (<http://data.gov.tw>)，可查到很多政府提供的開放資料，這些資料約一小時及更新一次，本練習要取用的式 JSON 格式，因此讀者若要自己練習其他資料來源，請找尋有提供 JSON 資料格式者。

空氣品質即時污染指標



資料集評分: 平均: 3 (1 位投票者)

[訂閱](#) [訂閱說明](#)

資料集描述	本資料集之舊有檔案下載連結 http://cdx.epa.gov.tw/CDX/OpenData/PSI.xml 將於10月31日移除，請使用本網頁新提供之檔案下載連結。環保署設於全國測站每小時發布之即時空氣品質監測資料；包括：空氣污染指標值（PSI）及各種污染物之小時濃度值
主要欄位說明	測站名稱(SiteName)、縣市(County)、空氣污染指標(PSI)、指標污染物(MajorPollutant)、狀態(Status)、二氧化硫濃度(SO2)、一氧化碳濃度(CO)、臭氧濃度(O3)、懸浮微粒濃度(PM10)、細懸浮微粒濃度(PM2.5)、二氧化氮濃度(NO2)、風速(WindSpeed)、風向(WindDirec)、發布時間(PublishTime)
資料資源	<div> CSV 檢視資料 </div> <div> JSON 檢視資料 </div> <div> XML 檢視資料 </div>

點選 JSON 格式，可檢視資料，所顯示的資料下載網址，即為本練習所要資料的來源網址。

資料資源

檢視資料資源

回到資料集

所有資料資源列表:

CSV	檢視資料
JSON	檢視資料
XML	檢視資料

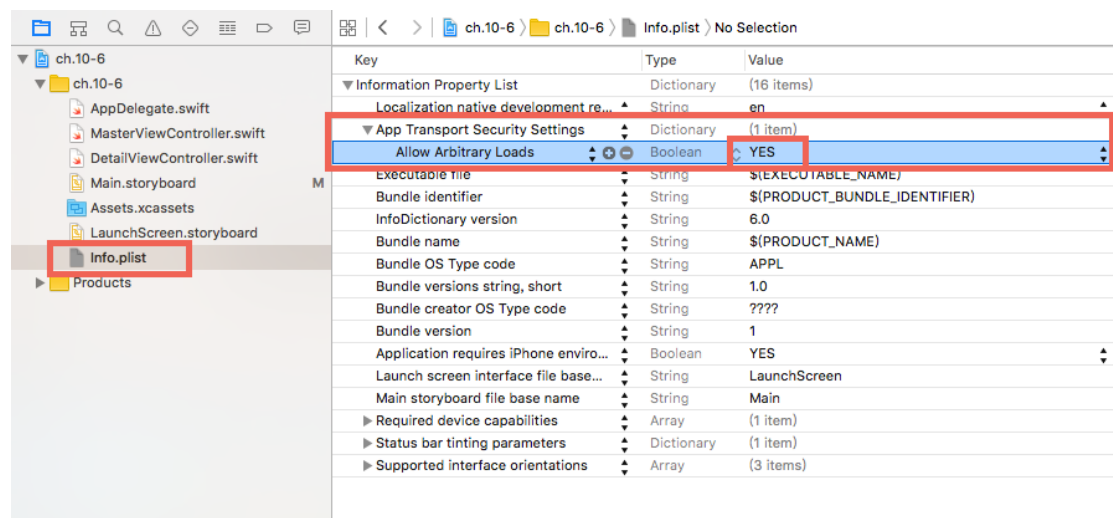
檔案格式	JSON
編碼格式	UTF-8;
資料下載網址	http://opendata.epa.gov.tw/ws/Data/AQX/?format=json
資料資源最後更新時間	103年11月11日 17:38:47

最佳瀏覽解析度為1024x768以上。國家發展委員會 地址:台北市10020中正區寶慶路3號 電話: 02-2316-5300 傳真電話: 02-2394-2165 隱私權保護、使用與資安政策
如對本站有任何問題或意見歡迎聯絡客服中心 電話: 02-2192-7111 信箱: egov@service.gov.tw © 2015 國家發展委員會 Rev:3

Step.3.

因目前 Xcode 要求使用安全性連線 https，但本次連接之網址為
<http://opendata.epa.gov.tw/ws/Data/AQX/?format=json>
因此需要修改 Info.plist 使其允許使用非安全性連線

開啟 Info.plist 於最上一列 Information Property List，點選旁邊的+號，進行新增一筆屬性值 App Transport Security Settings 項下新增 Allows Arbitrary Loads
於 Value 欄位，選擇值為 YES

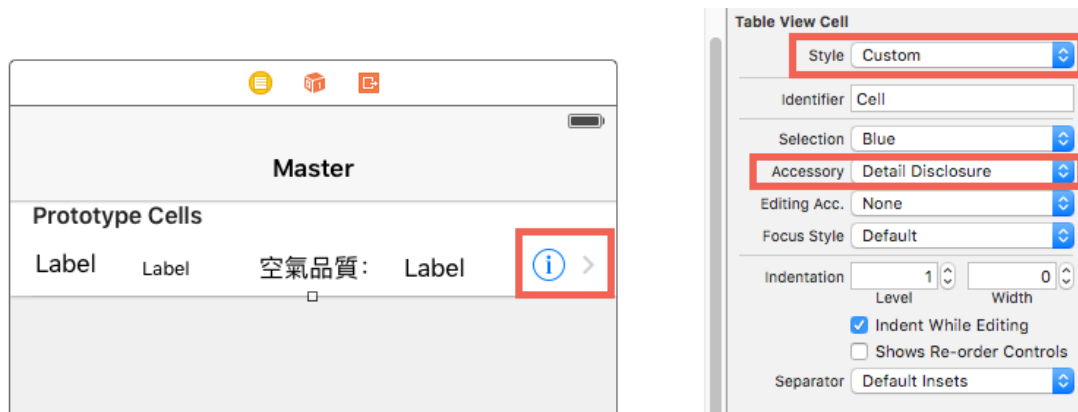


Step.4.

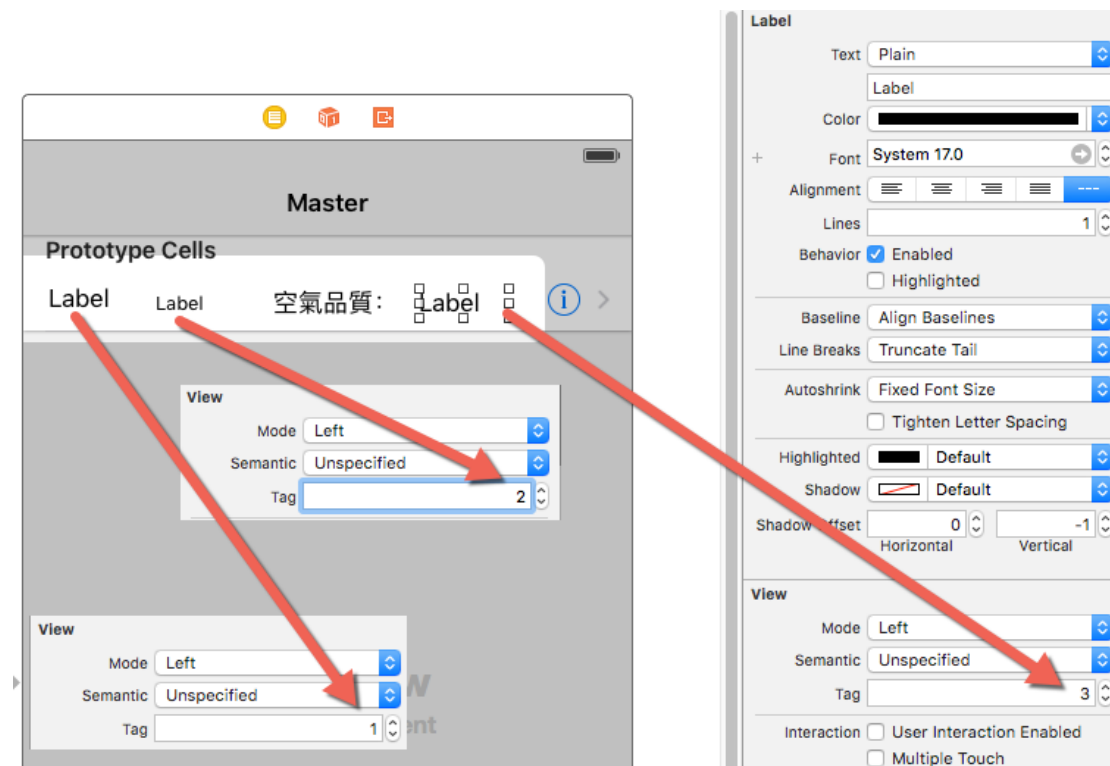
設計 Master 畫面

設定輔助選項為 Detail Disclosure ， 會顯示 ⓘ >

並新增四個 Label ， 將第三個 Label 文字改為 空氣品質：



三個 Label 分別給予 Tag 編號 1, 2, 3 。



Step.5.

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 myURL = NSURL(string: "http://opendata.epa.gov.tw/ws/Data/AQX/?format=json")

let jsonData = NSData(contentsOfURL: myURL!)

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`

改為

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

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

- (3) 使用 `viewWithTag` 方式將新增的三個 `Label`，分別命名為 `Label1`, `Label2`, `Label3`

```
let object = objects[indexPath.row] as! NSDictionary
let myLabel1 : UILabel = cell.viewWithTag(1) as! UILabel
let myLabel2 : UILabel = cell.viewWithTag(2) as! UILabel
let myLabel3 : UILabel = cell.viewWithTag(3) as! UILabel
```

- (4) 由 `NSDictionary` 中利用 `objectForKey(“ ”)` 的方式，取出 `County`, `SiteName`, `PSI` 三項數

值。

```
myLabel1.text = object.objectForKey("County") as! NSString as String
myLabel2.text = object.objectForKey("SiteName") as! NSString as String
myLabel3.text = object.objectForKey("PSI") as! NSString as String
```

(5) 由 NSDictionary 中利用 objectForKey(“”)的方式，取出 Status 數值，若為良好則顯示綠色，普通顯示藍色，否則則顯示紅色。

```
let myStatus : NSString = object.objectForKey("Status") as! NSString
```

```
if (myStatus == "良好"){
    myLabel3.textColor = UIColor.greenColor()
} else if(myStatus == "普通"){
    myLabel3.textColor = UIColor.blueColor()
} else {
    myLabel3.textColor = UIColor.redColor()
}
```

```
import UIKit

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: #selector
            (MasterViewController.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 myURL = NSURL(string: "http://opendata.epa.gov.tw/ws/Data/AQX/?format=json")
        let jsonData = NSData(contentsOfURL: myURL!)
        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
    let myLabel1 : UILabel = cell.viewWithTag(1) as! UILabel
    let myLabel2 : UILabel = cell.viewWithTag(2) as! UILabel
    let myLabel3 : UILabel = cell.viewWithTag(3) as! UILabel

    myLabel1.text = object.objectForKey("County") as! NSString as String
    myLabel2.text = object.objectForKey("SiteName") as! NSString as String
    myLabel3.text = object.objectForKey("PSI") as! NSString as String

    let myStatus : NSString = object.objectForKey("Status") as! NSString

    if (myStatus == "良好"){
        myLabel3.textColor = UIColor.greenColor()
    } else if(myStatus == "普通"){
        myLabel3.textColor = UIColor.blueColor()
    } else {
        myLabel3.textColor = UIColor.redColor()
    }

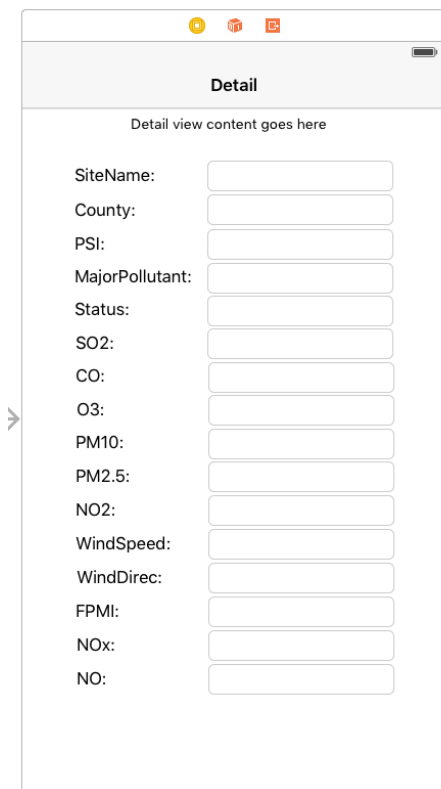
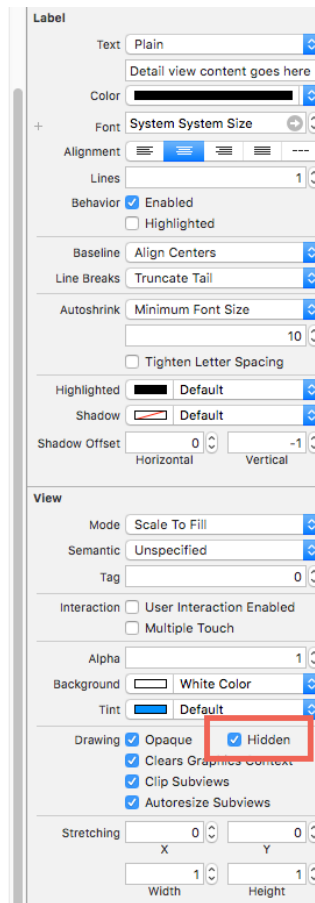
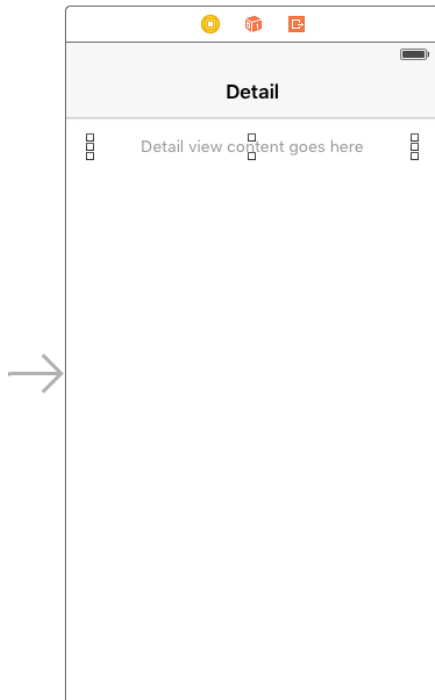
    return cell
}

```

Step.6.

開啟 Main.storyboard，於 Detail 畫面中，建置好明細資料呈現畫面。

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



```
import UIKit

class DetailViewController: UIViewController {

    @IBOutlet weak var detailDescriptionLabel: UILabel!
    @IBOutlet weak var t01: UITextField!
    @IBOutlet weak var t02: UITextField!
    @IBOutlet weak var t03: UITextField!
    @IBOutlet weak var t04: UITextField!
    @IBOutlet weak var t05: UITextField!
    @IBOutlet weak var t06: UITextField!
    @IBOutlet weak var t07: UITextField!
    @IBOutlet weak var t08: UITextField!
    @IBOutlet weak var t09: UITextField!
    @IBOutlet weak var t10: UITextField!
    @IBOutlet weak var t11: UITextField!
    @IBOutlet weak var t12: UITextField!
    @IBOutlet weak var t13: UITextField!
    @IBOutlet weak var t14: UITextField!
    @IBOutlet weak var t15: UITextField!
    @IBOutlet weak var t16: UITextField!

    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()
    }

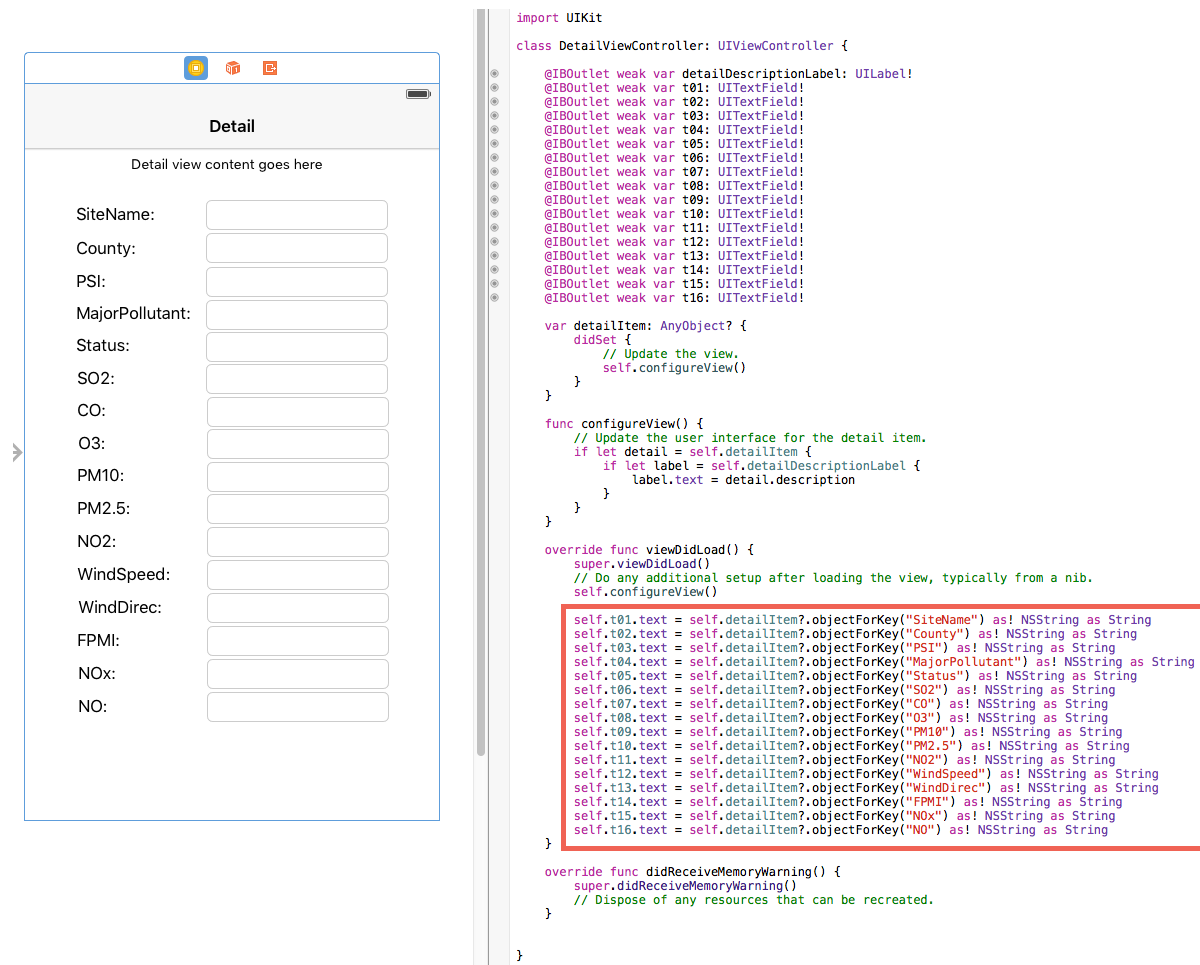
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

Step.7.

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

透過 `self.detailItem?.objectForKey("name")` 的方式取出 `NSDictionary` 中取出各項儲存值。

```
self.t01.text = self.detailItem?.objectForKey("SiteName") as! NSString as String
self.t02.text = self.detailItem?.objectForKey("County") as! NSString as String
self.t03.text = self.detailItem?.objectForKey("PSI") as! NSString as String
self.t04.text = self.detailItem?.objectForKey("MajorPollutant") as! NSString as String
self.t05.text = self.detailItem?.objectForKey("Status") as! NSString as String
self.t06.text = self.detailItem?.objectForKey("SO2") as! NSString as String
self.t07.text = self.detailItem?.objectForKey("CO") as! NSString as String
self.t08.text = self.detailItem?.objectForKey("O3") as! NSString as String
self.t09.text = self.detailItem?.objectForKey("PM10") as! NSString as String
self.t10.text = self.detailItem?.objectForKey("PM2.5") as! NSString as String
self.t11.text = self.detailItem?.objectForKey("NO2") as! NSString as String
self.t12.text = self.detailItem?.objectForKey("WindSpeed") as! NSString as String
self.t13.text = self.detailItem?.objectForKey("WindDirec") as! NSString as String
self.t14.text = self.detailItem?.objectForKey("FPMI") as! NSString as String
self.t15.text = self.detailItem?.objectForKey("NOx") as! NSString as String
self.t16.text = self.detailItem?.objectForKey("NO") as! NSString as String
```



Step.8

執行程式。

可以看到連線到即時空氣品質網站下載最新資料，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>>

Carrier 2:57 PM				
Edit		Master		+
基隆市	基隆	空氣品質:	35	i >
新北市	汐止	空氣品質:	41	i >
新北市	萬里	空氣品質:	40	i >
新北市	新店	空氣品質:	43	i >
新北市	土城	空氣品質:	45	i >
新北市	板橋	空氣品質:	54	i >
新北市	新莊	空氣品質:	41	i >
新北市	菜寮	空氣品質:	44	i >
新北市	林口	空氣品質:	50	i >
新北市	淡水	空氣品質:	43	i >
臺北市	士林	空氣品質:	37	i >
臺北市	中山	空氣品質:	47	i >
臺北市	萬華	空氣品質:	50	i >
臺北市	古亭	空氣品質:	45	i >
臺北市	松山	空氣品質:	49	i >
臺北市	大同	空氣品質:	54	i >
桃園市	桃園	空氣品質:	42	i >
桃園市	大園	空氣品質:	46	i >
桃園市	觀音	空氣品質:	46	i >

Carrier 2:57 PM	
Master	Detail
SiteName:	板橋
County:	新北市
PSI:	54
MajorPollutant:	懸浮微粒
Status:	普通
SO2:	4.7
CO:	0.57
O3:	18
PM10:	62
PM2.5:	35
NO2:	25
WindSpeed:	3
WindDirec:	276
FPMI:	3
NOx:	29.42
NO:	4.22