# Smart HMIS 開發手冊

Written by Andrew Wang. 2018/1/3

## 三大主軸

Smart HMIS 系統共有三大主軸，分別是手機APP，外網與內網。此三大主軸會在 Smart Hmis Project 的 Features 中加以說明並指派 Owner。

### Feature: 手機APP

手機APP的功能大致上與外網相同，不同在於外網是公開的網站，使用者可以瀏覽上面所有的網頁，如果要用到問卷調查或是下訂單的時候，才會需要進入LOGIN畫面，但是手機APP我們強迫使用者第一個動作就是登入，如果沒有帳號，就先行註冊。

這樣做的原因在於，會下載手機APP的使用者通常已經看過外網的資訊，或是對於健檢有需求，已經是我們所鎖定的客戶，因此，登入的原因在於蒐集客戶的資料並且直接引導客戶做問卷調查並且下單。

手機APP的功能大致分為下列幾項:

先是單獨頁面

1. Login

2. 如果忘記密碼，重設密碼

3. 如果沒有帳號，註冊

接下來進入 Portal 畫面，Portal 畫面則包含了以下的功能選單:

1. 健檢套餐瀏覽 (可以瀏覽所有套餐)

2. 檢前問卷調查 (問卷結果引導客戶選擇特定套餐)

3. 線上下單預約 (直接線上預約檢查醫院，日期，並下訂單付款)

4. 個人健康中心 (體檢歷史紀錄，電子體檢報告，分析，建議，追蹤)

5. 個人信息維護 (設定)

目前APP功能只包含了檢前與檢後，檢中目前不確定是否適合用手機通知現場狀況，實際情況是在醫院會有電子報到系統

Smart HMIS Project Feature

Written by Andrew. 2018/1/3

### Feature: 外網

外網的功能大致上與手機APP相同，不同在於外網是公開的網站，使用者可以瀏覽上面所有的網頁，如果要用到問卷調查或是下訂單的時候，才會需要進入LOGIN畫面。

因此，外網會有一個入口網站，就如同每家公司的網站一樣。

Portal 畫面則包含了以下的功能選單:

1. 首頁(裡面包含第一眼希望使用者看到的東西)
2. 健檢套餐瀏覽 (可以瀏覽所有套餐)
3. 檢前問卷調查 (問卷結果引導客戶選擇特定套餐)
4. 線上下單預約 (直接線上預約檢查醫院，日期，並下訂單付款)
5. 個人健康中心 (體檢歷史紀錄，電子體檢報告，分析，建議，追蹤)
6. 關於我們 (公司簡介)
7. 個人信息維護 (設定)

另外可以在 Top Bar 上面有兩個小連結

1. 登入
2. 註冊
3. 聯絡我們 (類似留言版功能)

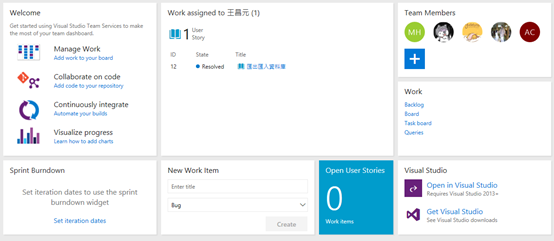
Smart HMIS Project Feature

Written by Andrew. 2018/1/3

### Feature: 內網

內網的功能則是完全不同的概念，其使用者為工作人員包含業務、醫護人員、以及檢後服務人員，內網必須經過LOGIN之後才能使用。因此，內網在一進入的時候，就會出現一個 LOGIN的畫面。

在LOGIN 之後，就會進入內網的Portal畫面，也就是內網的主頁，同樣的，這個主畫面可以設計成一個 Dashboard，讓每個人可以看到今天該注意的事項，如下面的範例。



內網包含了以下的功能選單:

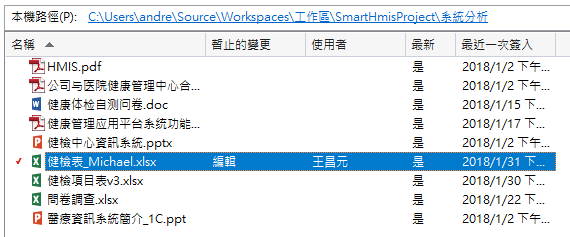
1. 首頁(裡面包含第一眼希望使用者看到的東西)
2. 健檢套餐瀏覽 (可以瀏覽所有套餐)
3. 檢前問卷調查 (問卷結果引導客戶選擇特定套餐)
4. 現場下單繳費 (直接下訂單付款)
5. 報到(客戶報到，報到機印出體檢單，讓客戶帶著單子跑站點)
6. 站點狀態(只是觀察站點排隊情形，客戶在站點門口刷條碼排隊時，站點機會提供建議)
7. 站點資料輸入(人工或自動，含醫囑建議)
8. 客戶健康管理 (體檢歷史紀錄，電子體檢報告，分析，建議，追蹤)
9. 單位數據分析 (分析單位的員工健康統計分析)

以上為檢中檢後的功能列表。

#### 健檢套餐

內網最重要的工作之一，就是健檢套餐的設計，我們的套餐是參考慈銘體檢中心的資料而來，原始系統分析資料存放於 健檢項目表v3.xlsx 檔案中，由我負責整理，新套餐的擴充則放在 健檢表\_Michael.xlsx 檔案中，資料由 Michael 蒐集，最後再由我完成整理。

健檢套餐的資料維護功能，則由 Alex 負責，預計2/2 完成，相關資料庫的詳細說明，則放在 Wiki 三之三 以後的說明文件。如下圖：



檔案內容如下圖：

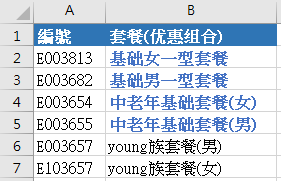


前三個TAB是編號的紀錄，由於編號不可以重複，所以特別整理出三個表格，有新增編號的時候，也會加入此三個表格。

從第四個TAB開始，就是套餐的內容，我已經將套餐展示方式用大綱式，分成兩層，第一層是套餐裡面的大項目(我稱之為 Level 1, 也就是 type\_id = 1)，第二層則是大項目裡面的小項目(Level2, type\_id = 2)。

##### 編號

健檢套餐裡面每一層都有編號，L0編號表就是套餐(combo)的編號，如下圖：



套餐編號是以E開頭(當然，這些原則可以改)，後面有六碼數字，其實我們在設計階段是可以隨便編，數字沒有原則，只要不重複就好，我們從慈銘體檢下載的套餐，會發現男生與女生共用一個編號，但是我的原則是必須不同，最簡單的方式就是最前面的數字加一個1，例如young族套餐。

L1編碼表是大項目的編碼表，通常是指可以被套餐選取的編碼表，如下圖：



此表的編碼原則是三個英文字母開頭，後面接兩個數字，最後面可能還會加個文字，當數字快要用完的時候，也可以由兩碼改為三碼，所以這個部分的編號我並沒有規定長度，長度可能是5或是6。

在這理會有一些項目是 “單項” 的，這是因為這些單項的項目並沒有子項目，是單獨存在的，而這些單項項目的編號，則是兩個英文字母加上三個數字，與前面說的編號原則不同，這是因為單項用的是小項目的編碼原則。

單項項目的定義是，它是一個小項目，但是卻可以和其他大項目一樣被套餐選取。依照這個定義，我們就知道這些單項項目在資料表裡面的 type\_id = 2。

L2編碼表則是小項目，也就是體檢的細項內容，如下圖：



這個部分的編碼就很多了，編碼原則是前面一或兩個英文字母，再加上後面三到四個數字，目前我只用到最多兩個字母與三個數字，如果不夠用，數字部分還可以變成四個，我建議這裡不要將前面的英文字母增加為三個，因為這樣會和 L1 的編碼混淆。

##### 套餐

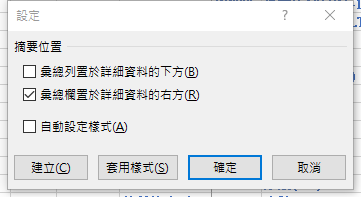


如上圖，目前已經完成的套餐有六個，分別是基礎男女，中老年男女，young族男女，別太執著女男或是男女的順序，前兩組我是用女男的順序，後面就是男女的順序，如果想在網頁中顯示統一的順序，用 order by 即可。

最左邊的 “+” 符號可以讓你展開大綱，看到裡面的細項，這是 Excel 的技巧，方便我們觀察。至於 “+” 要在上面還是下面，按下 “大綱” 右下角的小箭頭符號，如下圖，就可以調整：



第一個選項不要打勾即可，如下圖：為何我要特別說明這個部分？是因為很容易忘記，在此提醒。



在 Excel 表格裡面有一個 “pro\_id” 欄位，其實指的就是 product 資料表裡面的 product\_id 欄位，這是該項目在範本資料庫中實際的 id，這個數字是遞增產生的，我們不能控制，將它記錄於此，在我們查看實際資料庫中的資料表裡面的資料的時候會比較方便。

由於這裡的檢查項目是不可以重複的，所以我特別保留了 eng\_name，因為在醫學上使用的英文縮寫是標準的且大部分不會重複的，另外也擔心會有不同的中文翻譯或是不同的兩岸用語。

這裡比較複雜的地方就是單位，單位的整理並沒有放在這裡，這個部分會在系統開發的後期才會與專業人士討論。

##### 優惠價格

在每個套餐的左上角會記錄價格，包含了市場價格與優惠價格，這些資訊都會被記錄到資料庫中。

##### 分工

在設計階段，套餐的蒐集由 Michael 負責，從慈銘體檢將套餐資料輸入 Excel 表中，並將已存在的編號輸入完成。

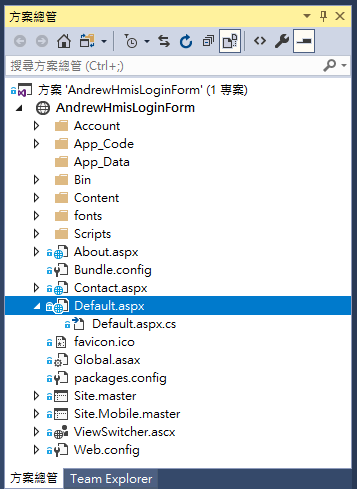
新的未編號項目則由我負責整理，整理好的套餐我會將TAB名稱加上 “(完成)”。

程式設計部分由 Alex 負責。

## 操作手冊

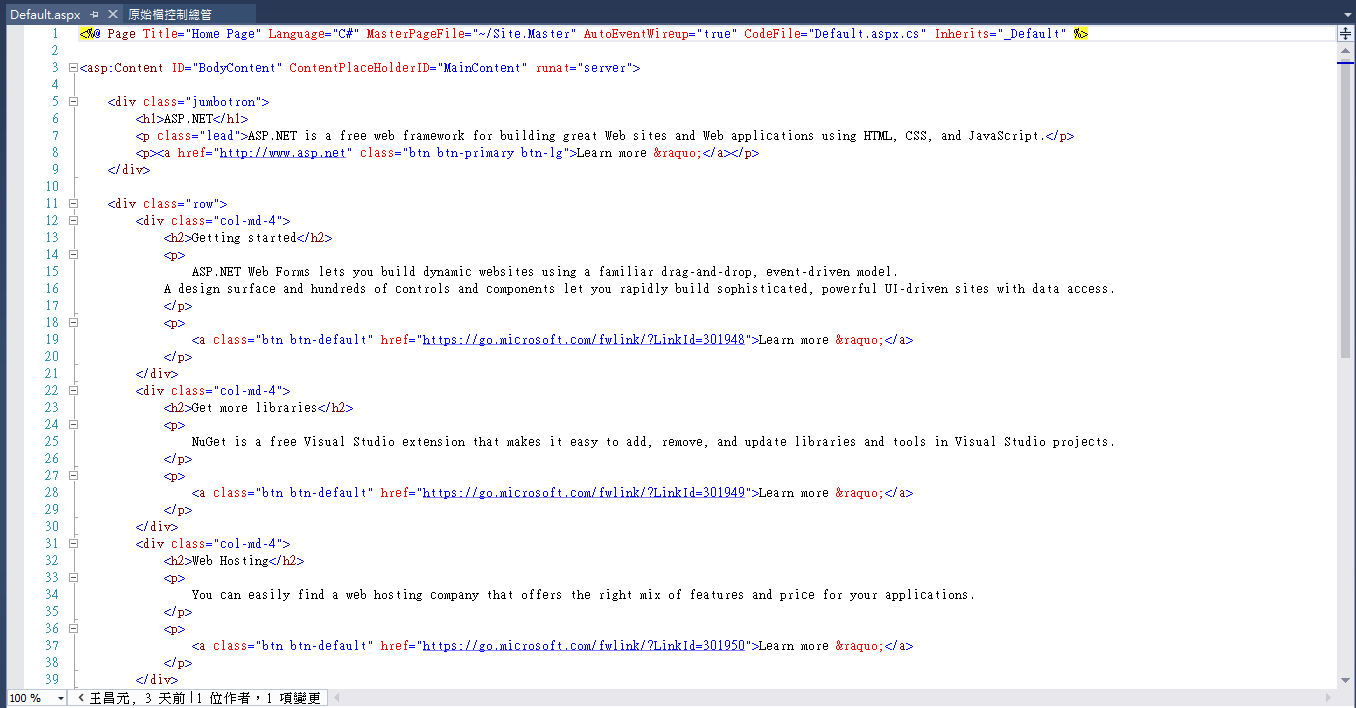
### 一、看懂第一個 Web Form Application

當我們新建一個 WebForm 方案的時候，VS會透過內建的 Template 建立一個半成品，如下圖：

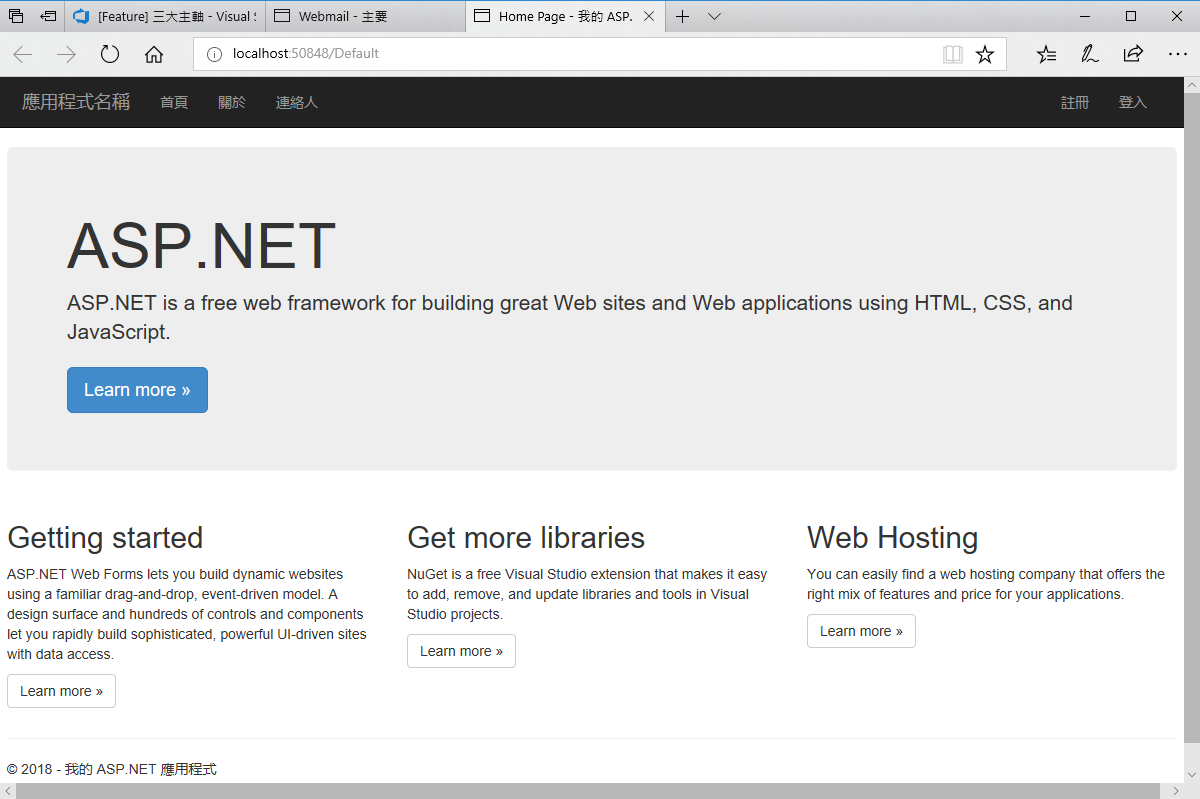


首先，我們先弄清楚這個程式的架構。先找到進入點吧，既然我們建立的是一個網站，那麼網站的進入點不是 index 就是 Default，這裡我們一眼就可以看到 Default.aspx，這就是網站的進入點。

開啟此檔案，可以看到下圖：



執行此檔案，結果如下：

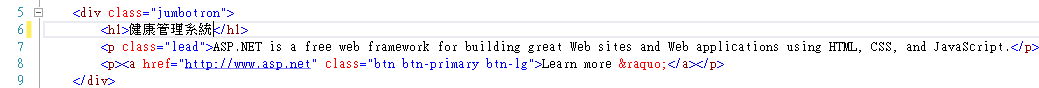


這個模板，其實就是我們想要的，裡面包含了首頁、關於、聯絡人、註冊以及登入。我們可以藉由修改這個模板來完成我們的健康管理系統。

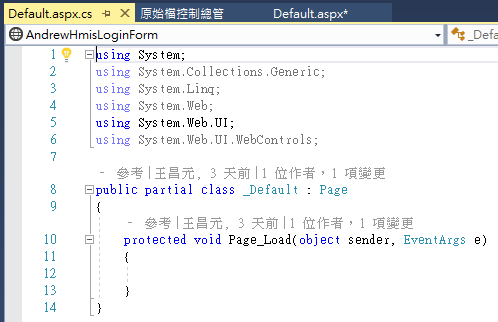
現在，我們從頭看一下程式架構。一般我們製作首頁的時候，通常是存成 Default.html 檔案，但是這裡為何是 Default.aspx檔案呢? 而在這個 aspx 檔案的裡面，為何還有一個 Default.aspx.cs 檔案呢?

理由很簡單，html 是靜態網頁，而 WebForm 是 Active Server Page 的動態網頁，跟php的概念一樣，aspx在 Server端執行之後才將結果 html 傳到 User 的瀏覽器上。

我們可以直接修改 aspx 的原始檔案來改變網頁的內容：

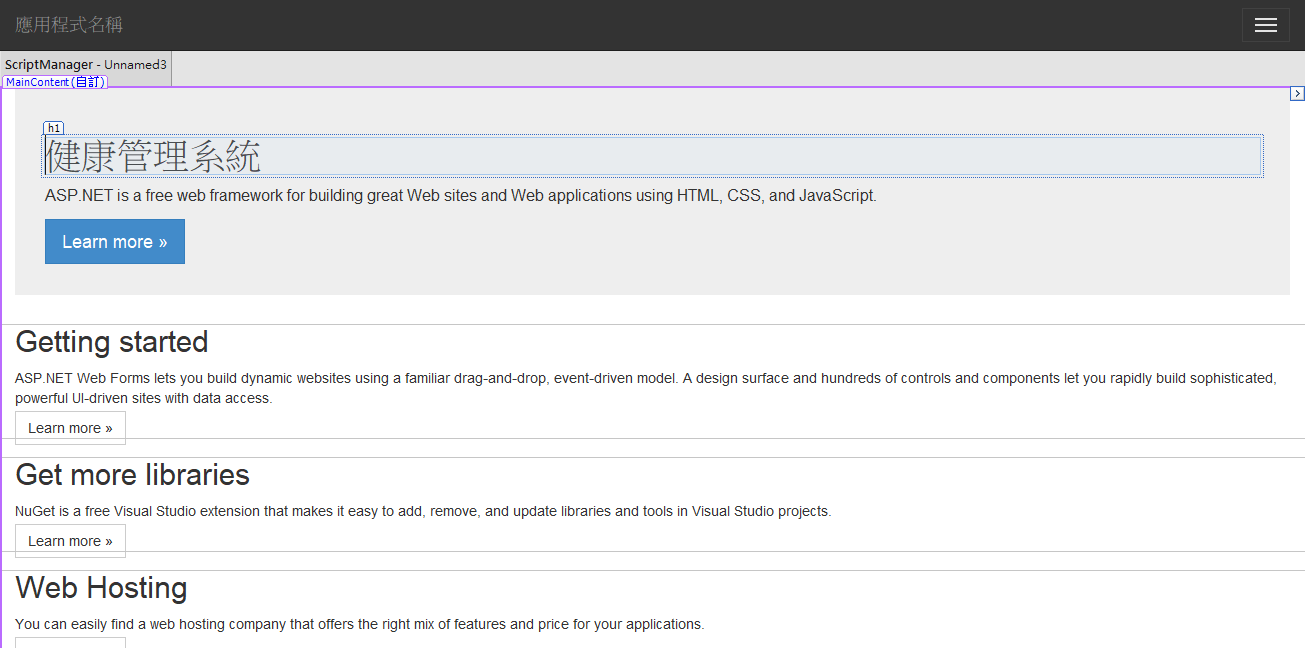


那麼，Default.aspx.cs 就是 Default.aspx 的 C# 程式碼了。內容如下：

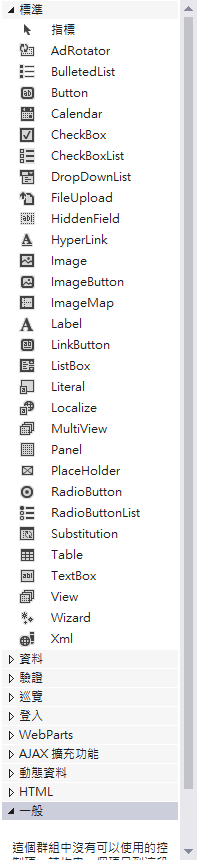


這裡是處理事件的 C# 程式碼，目前只有一個空的 Page\_Load 事件處理在裡面。當我們有新的事件要處理的時候，C#程式碼就可以寫在這裡，例如開啟資料庫。

在回頭看 Default.aspx，要修改內容，可以直接修改裡面的原始碼，也可以進入”設計”畫面，我們點一下左下角的[設計]，即可進入設計畫面如下：

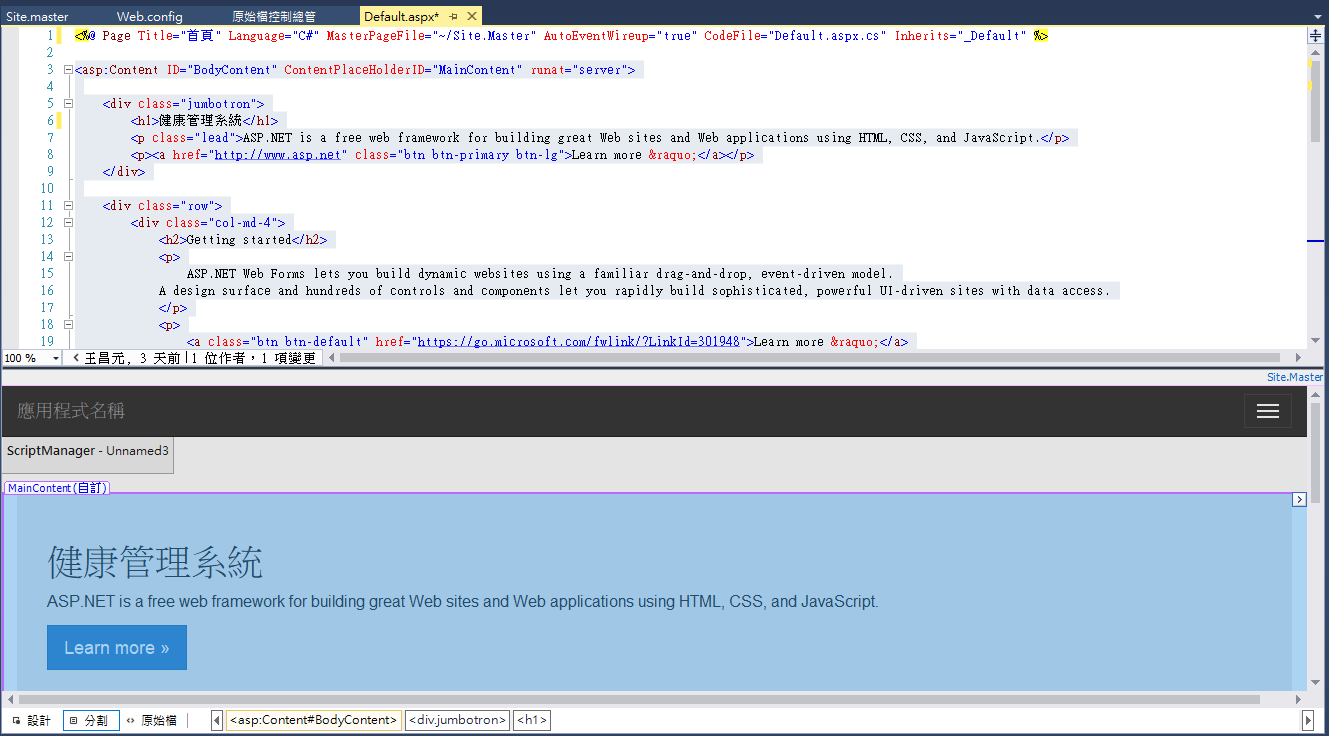


當進入設計畫面之後，左邊的工具箱就會顯示出來了：



透過這個工具箱，我們就可以如同在 Thingworx 裡面使用 Widget 一樣來編寫程式了。

如果我們再點一下[分割]，就會看到下圖：

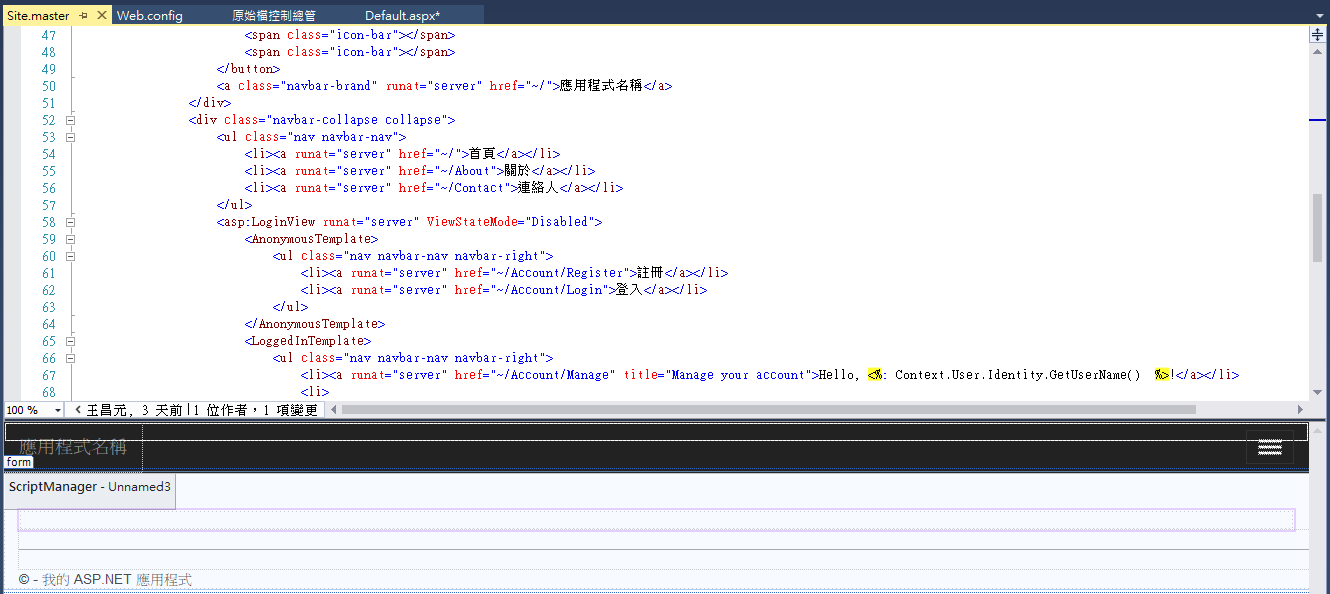


從這個畫面中，我們就可以很清楚的知道，Default.aspx 的內容是存放在 MainContent 裡面的，我們能夠編輯的部分，也就在 MainContent 這個區域裡面，當滑鼠移動到這個區域外面的時候，就會出現紅色禁止標誌，代表那些區域是無法在此編輯的。

我們先看第一行:

<%@ Page Title="首頁" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

透過執行首頁的畫面我們知道，在首頁內容的上方，有一個Menu Bar，裡面有註冊、登入等選單，這個 Bar 會出現在每一頁裡面，定義在 MasterPageFile="~/Site.Master" 中，因此，如果要修改選單，就必須打開 Site.Master檔案：



對應一下我們過去使用 Thingworx 的經驗，Default.aspx, About.aspx 這些就是 Mashup，而 Site.Master 就是 Master Mashup。只要在 Mashup 中指定了 Master，執行Mashup 的時候 Master 就會套在外面，成為固定在上方或是左方的選單。

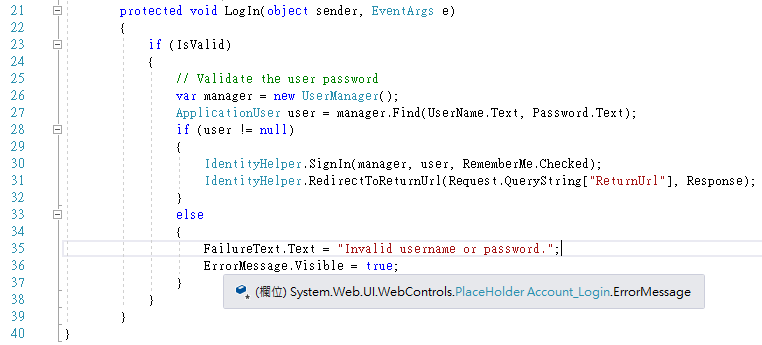
由於 Visual Studio 讓你看到所有的程式碼，所以編輯的時候要特別小心，不要隨意變動系統自動產生的程式碼。以免程式無法執行。

我們目前要寫的第一個畫面，是 Login 畫面，按照這個模板的架構，請打開 Account 底下的 Login.aspx。

在登入畫面裡面，最重要的部分就是[登入]按鈕，在 Login.aspx 程式碼中，我們可以看到這段：

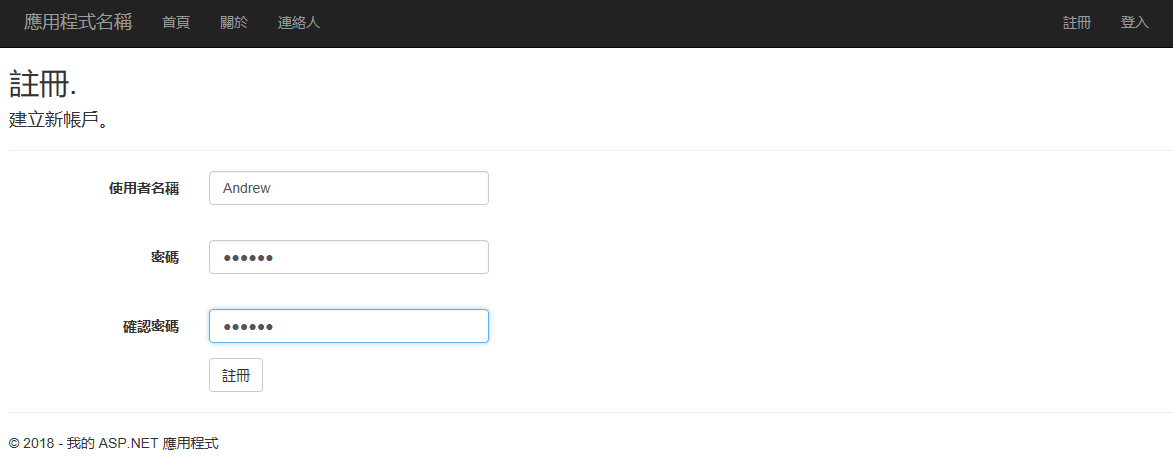


<asp:Button runat="server" OnClick="LogIn" Text="登入" CssClass="btn btn-default" />，當按鈕被按下的時候，會觸動 OnClick 事件，事件處理的名稱是 LogIn，C# 程式碼則寫在 Login.aspx.cs 裡面。

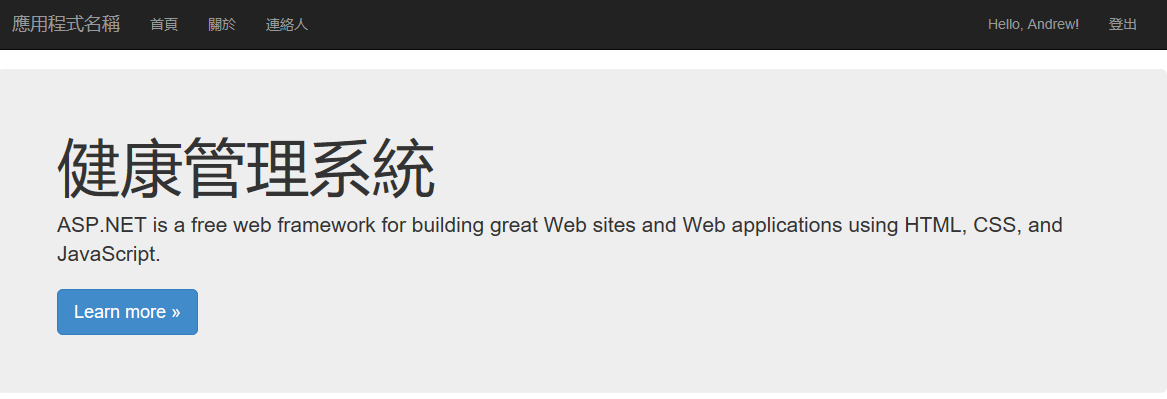


也就是說，如果我們要修改 VS 的內建登入方式，就必須在這裡修改程式碼。

要對這個程式做修改，還不是那麼容易，我們先看一下它的運作方式。首先進入首頁，然後按下右上方的[註冊]按鈕，則進入註冊畫面，輸入註冊資料：



我自行輸入 Andrew (帳號密碼相同)，則就可以直接登入了，右上方出現了 Hello Andrew以及[登出]按鈕：



微軟到底用了什麼方式來處理註冊以及登入登出? 請先參考下面的網址：

<https://youtu.be/9l7AM5R9yQQ>

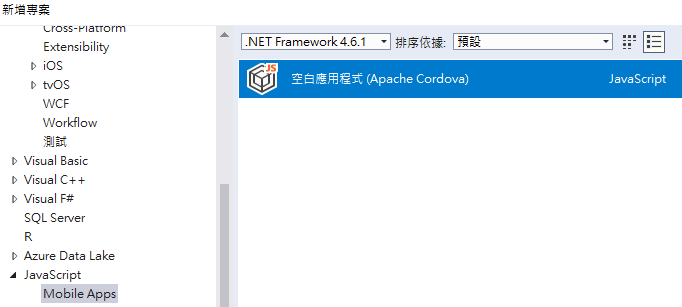
花點時間研究清楚，我們才能繼續往下走。

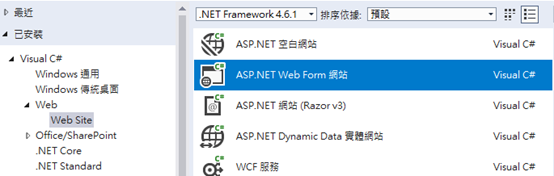
#### 一之一、建立新的Visual Studio方案

三大主軸之下，我們有兩種開發模式，第一種是手機APP，第二種就是 Web Application。

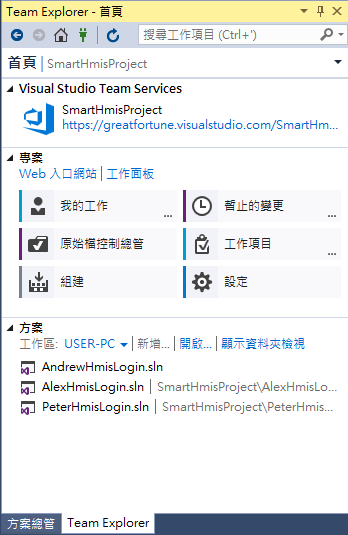
在Visual Studio 裡面建立一個新方案有非常多種模式，在這個健檢管理系統的案子裡面，我們決定手機APP用 jQuery 的技術開發，而內網與外網則使用 Web Form 的技術開發，所以，選擇方案 Template 的說明如下：

**手機APP:**  
無論是 Android 或是 iOS，都有一個一樣的模板，叫做 WebView應用程式，如下圖，所以，可以在Visual C# / Android 或是 Visual C# / iOS / iPhone 裡面找到。



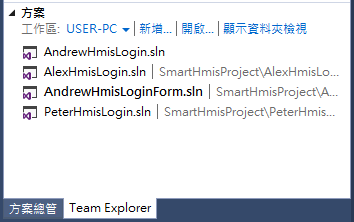
**內網／外網：**  
可以在 Visual C# / Web / Web Site 裡面找到 ASPNET Web Form 網站。  
  
先確定好要開發的方案之後，我們就以Web Form為例子，建立一個新方案。

**步驟一**

  
在 Team Explorer 的工作區中，確定已經連結了 SmartHmisProject，然後在下面的[方案]中點選[新增]按鈕。

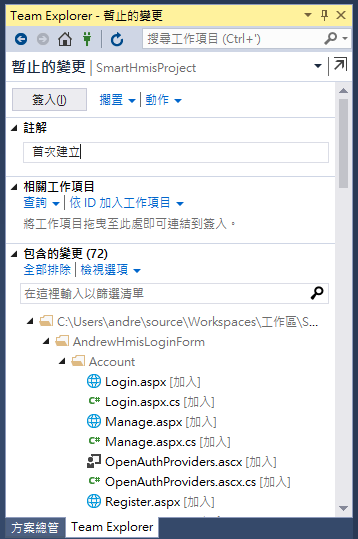
**步驟二**  
  
出現新增專案按鈕之後，請先按照前面的說明選擇專案種類，然後就是命名，因為我們是五人共同開發，所以駝峰式命名原則是英文名字、專案名稱、功能名稱。然後就是位置，經過幾天的研究測試，我建議直接選擇我們 Team Service Project 的目錄，就是”工作區”底下的 “SmartHmisProject”，最後，右邊兩個 checkbox 務必要打勾(請再三檢查)，就可以按下[確定]鍵了。

**步驟三**  
  
接下來請耐心等待一下，系統會花點時間跑模板，跑完之後，專案會出現在 Team Explorer 裡面，如下圖。



這時候，專案只是在我們的本機端建立完成，還沒有上傳到 Team Server，我們必須將其簽入才行。

點一下[暫止的變更]，會在包含的變更底下，看到剛剛建立出來的方案的所有檔案，如下圖：

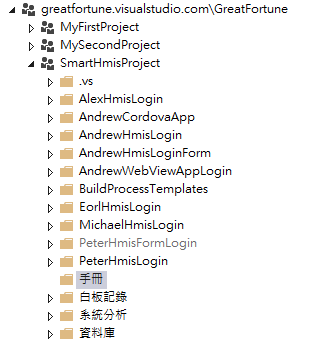


簽入之前，最好先去 “排除的變更” 裡面看一下，如果有"偵測到xxx項目加入" 請先把這些項目都升級。然後，請在上面的註解處，寫上這次變更的註解(建議寫一下，在這裡也許可以不用寫，但是如果是 Git 則一定要寫)，然後按下[簽入]。

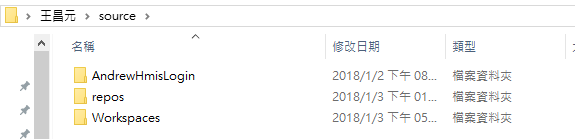
最後出現成功簽入就代表成功了，方案已經在本機端與 Team Server 端各有一份了。

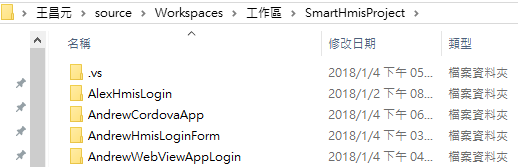
#### 一之二、刪除現有方案

由於我們是團隊開發，每個人新增的方案都會出現在”原始檔控制總管”裡面，如下圖:



當然，我們的原則是，別人新增的方案我們不要去刪除，只處理自己的，這樣比較不會出錯。以目前的例子而言，由我創建的方案就有四個，在 Team Server 裡面，都是存放在 SmartHmisProject 這個團隊專案裡面 (PS: 這裡講的團隊專案 Project 與方案 Solution要分清楚，團隊專案指的是 Team Service 裡面的專案，而方案則是指 Visual Studio 裡面的。另外， Visual Studio 的方案裡面，還可以建立專案 Project，請不要跟團隊專案搞混了)，然後我們再看一下它們實際存放的位置：



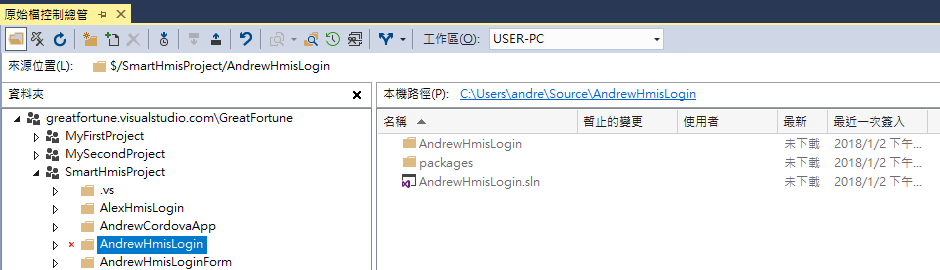


大家可以發現，AndrewHmisLogin 是存放在 source 目錄底下，而其他的三個，則是存放在 source\Workspace\工作區\SmartHmisProject 目錄底下。

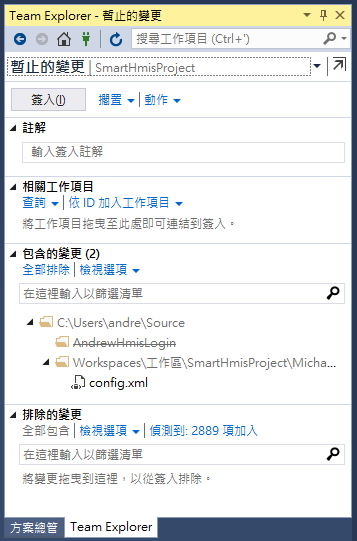
為何會如此? 這是因為我在建立方案的時候，所選擇的位置不一樣，建立方案時，其實可以放在任何位置，但是為了不讓自己混淆，我在上一篇的分享中，建議放在與團隊專案一樣的目錄底下，也就是 SmartHmisProject，不然就會跟我一樣，VS方案放在兩個地方，自找麻煩。

為了解決亂象，我決定將 AndrewHmisLogin 方案刪除，刪除的方式很多，但是建議不要把本地端的目錄直接刪掉，既然我們是 Teamwork，我們就從 Team Service 裡面來刪除它。

我們回到”原始檔控制總管”，在要刪除的 AndrewHmisLogin 上面按滑鼠右鍵，再選擇[刪除]，這時，方案就被刪除了，如下圖：

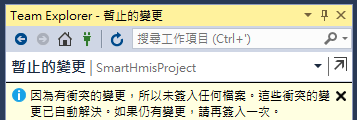


這時候，其實只有你自己看見紅色叉叉，並沒有同步到 Team Service 中，接下來，請到右邊的 Team Explorer 中，查看”暫止的變更”，就會看到我剛剛所做的變更，如下圖：



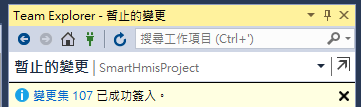
在包含的變更中，出現了 AndrewHmisLogin，並且有刪除線，(PS: 下面還有一個 Michael 的變更，是明雨的變更，跟此例子無關)我們只要按下[簽入]按鈕就可以了。

接下來就會出現一個非常重要的狀況，如下圖：



看到這個訊息，千萬不要不理它，這代表衝突已經自動解決，必須再按一次[簽入]按鈕。

請切記，一定要看到下面的畫面才是簽入成功：



成功之後，我們再到”原始檔控制總管”查看，就會發現原本打紅色叉叉的方案不見了。就代表說，在 Team Service 裡面，這個方案已經不見了，所有的團員只要在同步之後，就看不見此方案。

但是，在我的本機端的目錄裡面，方案還是存在的，如下圖：



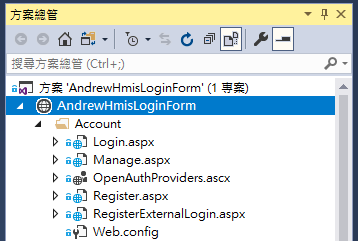
理由很簡單，我在”原始檔控制總管”只是代表我不要將此專案放在 Team Service 裡面，並不代表我不會在本機端繼續開發。如果我真的不想要此方案了，我就會自行刪除目錄。

Written by Andrew Wang. 2018/1/8

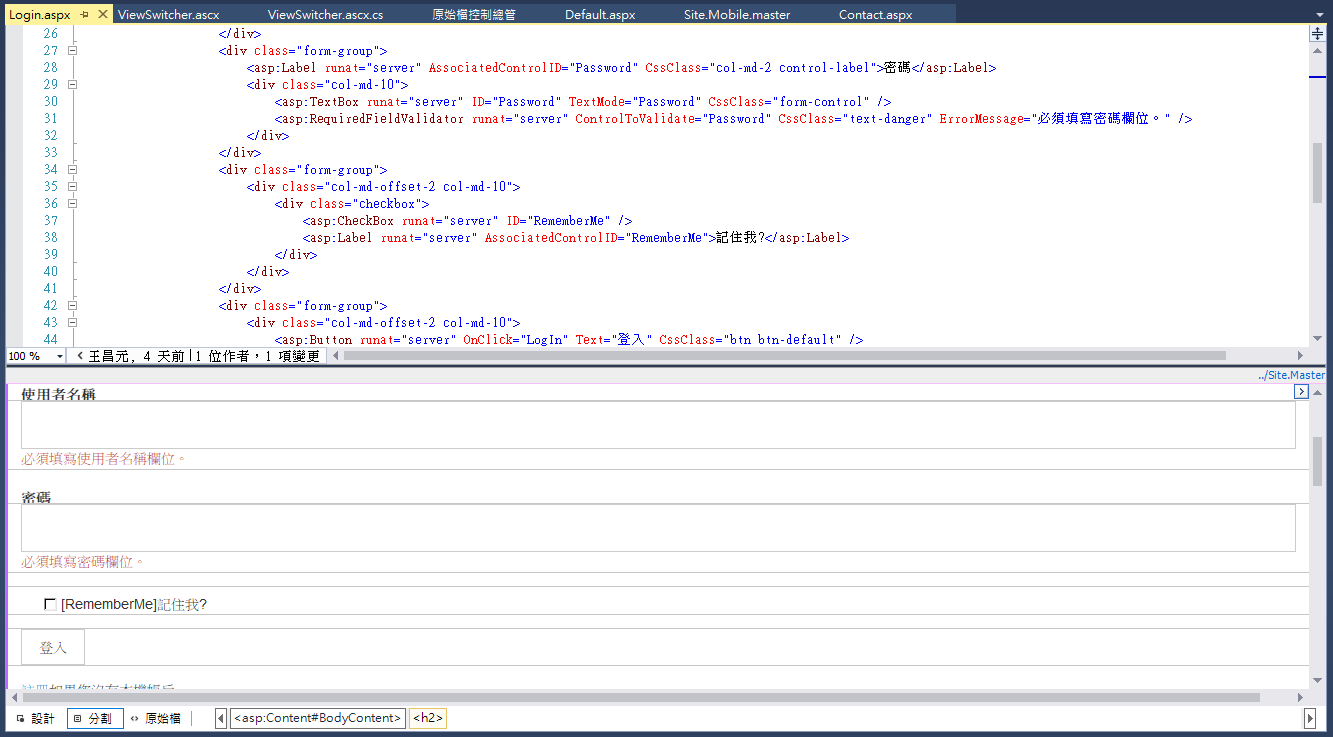
#### 一之三、WebForm 內建的使用者管理功能

很幸運的，Visual Studio 的 WebForm模塊本身就含有只用者管理功能，其中包含了註冊、登入與修改密碼。

所有的使用者管理功能，都放在下面的目錄：



##### Login.aspx



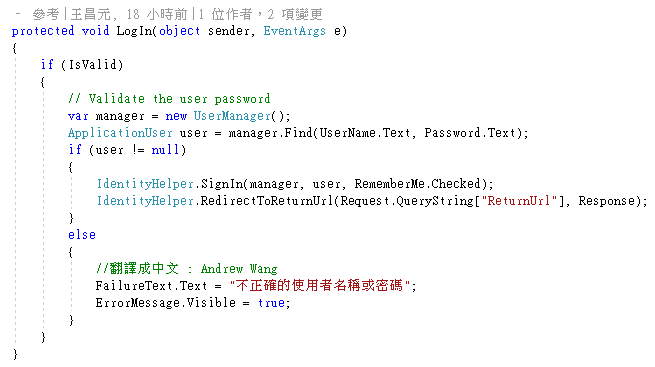
Login的功能很簡單，就是兩個 Textbox 加上一個登入按鈕，按下登入按鈕，則執行 LogIn 程序。

<asp:Button runat="server" OnClick="LogIn" Text="登入" CssClass="btn btn-default" />

LogIn 則寫在 LogIn.aspx.cs 裡面：

程式的第一行是 using Microsoft.AspNet.Identity; ，代表這個程式引用的 Microsoft 內建的程式庫 AspNet.Identity，這是處理標準使用者管理的程式庫，後面出現的類別與方法，都源自於這個程式庫。

接下來請看 LogIn 這個方法：



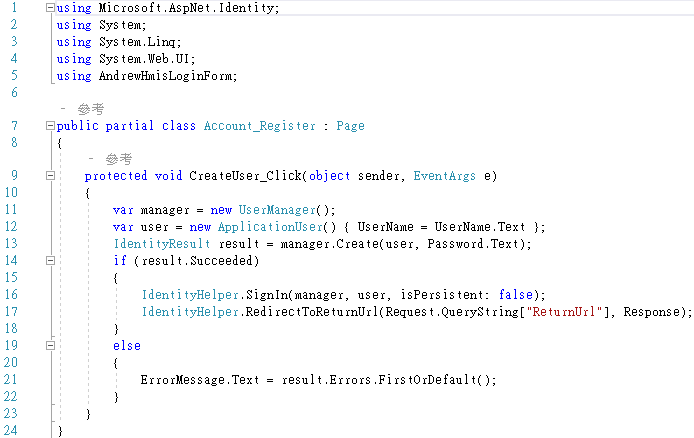
首先宣告一個 manager 物件為 UserManager() 類別，然後用 manager.Find 方法傳入帳號跟密碼，並將結果傳給一個屬於 ApplicationUser 的物件 user，接著檢查 user 的結果，如果不是 Null，代表此 user 存在，就直接執行 IdentityHelper類別中的 SignIn 方法，完成 LogIn 並切回原來的畫面。

這些都是標準的寫法，所以我們幾乎可以不用做任何修改。

##### Register.aspx



同樣的，如果使用者不存在，可以進入註冊畫面進行註冊， OnClick = "CreateUser\_Click" 表示當按下註冊按鈕的事件產生，則執行 CreateUsser\_Click。打開 Register.aspx.cs 便可以看到程式碼：



透過 var user = new ApplicationUser() { UserName = UserName.Text }; 建構一個 user 物件並透過建構子先將 UserName 傳入。

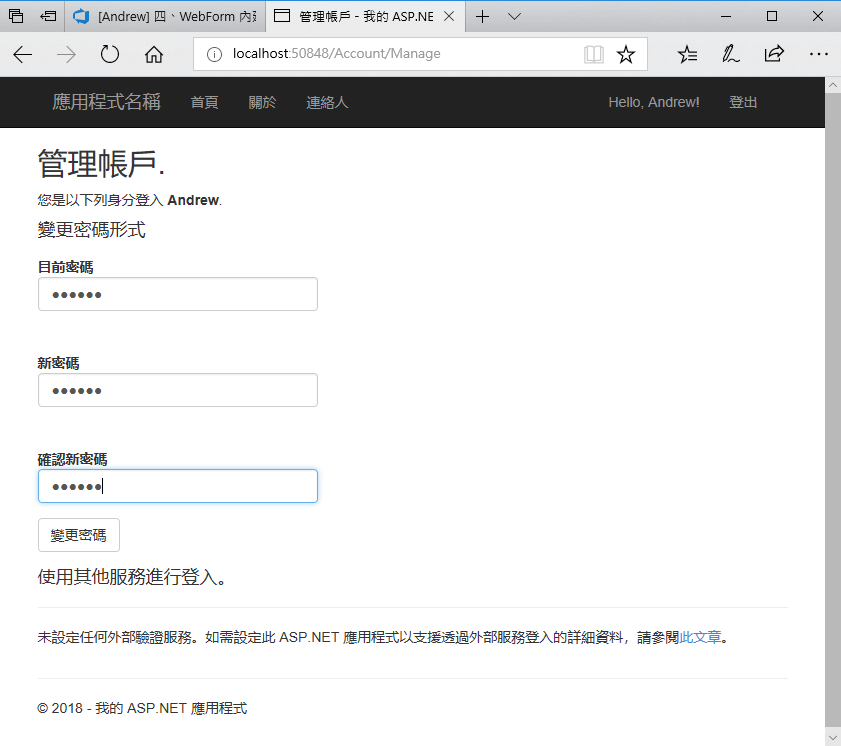
IdentityResult result = manager.Create(user, Password.Text); 直接 Create 一個 user。

##### Manage.aspx

管理功能就只有變更密碼，程式碼如下圖：

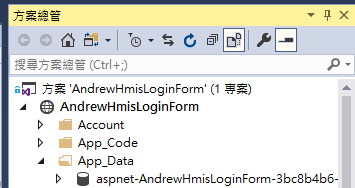


這裡提供了兩個方法，一開始會先檢查是否有密碼，如果沒有就直接SetPassword，如果有，則要 ChangePassword。



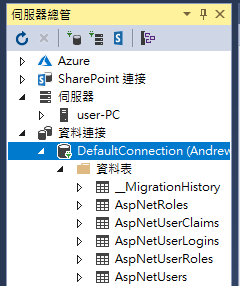
##### App\_Data

我已經在測試頁面中註冊了一個 Andrew 的帳號，然而這個資料到底存在哪裡？請打開 App\_Data。

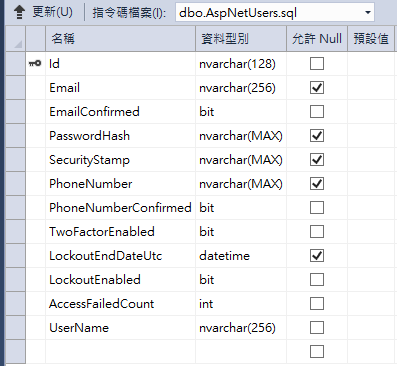


我們會發現這裡多了一個名為 SQL Database，名為： 

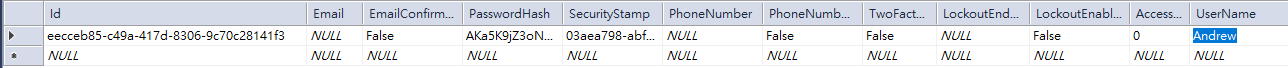
我們來看一下這個資料庫的內容。只要在 App\_Data中的資料庫上面點兩下，就會進入伺服器總管：



打開 AspNetUsers 就可以看見其資料欄位：



或按下右鍵選取[顯示資料表內容]：

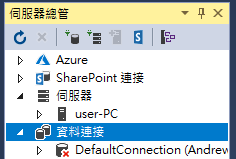


到這一步我們就有兩種思考模式，一是將此資料庫整併到我們的資料庫裏面，另一個就是保留此內建的資料庫，而將我們的 customer 資料表加一個 ID 欄位儲存這裡的 ID，eecceb85-c49a-417d-8306-9c70c28141f3。此處的ID是自行產生的 GUID，保證唯一。

#### 一之四、WebForm連結 hmis 資料庫

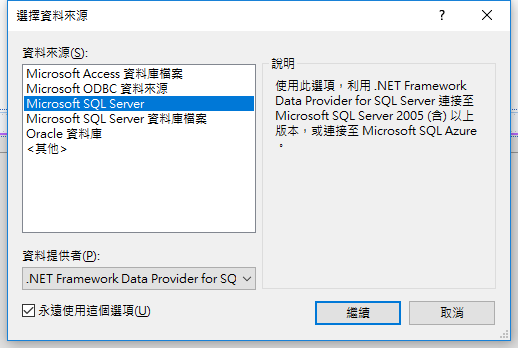
接下來，我們試著把 hmis 資料庫連結到我們的 WebForm，並顯示 customer 資料表在 contact.aspx 頁面中。

我們可以使用 Visual Studio 的伺服器總管來連接並管理資料庫，無須透過 SQL Server Management Studio來做，這樣可以少學一樣工具。所以第一步我們就來連結 hmis資料庫，以確定我們的資料庫連結是沒有問題的。



打開伺服器總管，我們可以看到自己的伺服器，我目前這台叫做 user-PC，伺服器的下面，就是資料連結，其實就是要建立一個 Connection到 user-PC 的 SQL Server。

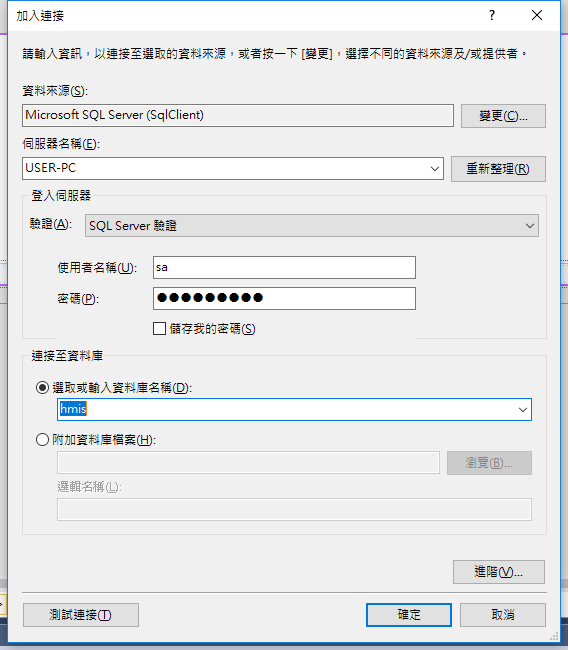
在[資料連接]上面按滑鼠右鍵選取[加入連接]，就會出現下面的對話框：



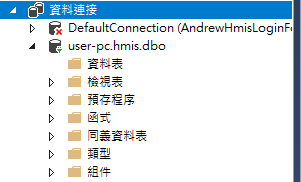
我們選擇 Microsoft SQL Server，這應該是內建的，無須安裝，這就是我為何選擇 SQL Server 而不用之前的 MySQL 的原因，微軟的開發工具對自家的資料庫的相容性總是比較好一點。

按下[繼續]之後，就會出現下面的對話框讓我們輸入SQL Server 的資料。

伺服器名稱的地方，理論上可以用下拉選單選到，但是如果下拉選單裡面沒有選項也不用擔心，我們可以自行輸入。接下來請參考下圖：



連結之後，就會看見 hmis 出現在資料連接下方。

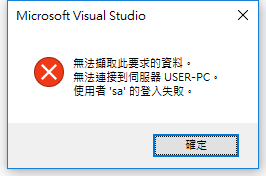


連接之後，會在屬性裡面出現連接字串：

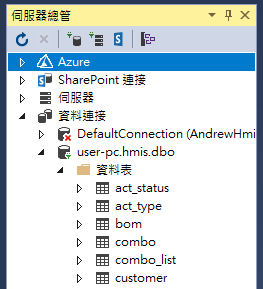
Data Source=USER-PC;Initial Catalog=hmis;User ID=sa;Password=\*\*\*\*\*\*\*\*\*\*\*

這點我們先保留，後面有機會用到。

PS: 如果在這裡出現這樣的問題，就是點開資料表之後會出現錯誤如下, 目前我查過 Google，無法解決。唯一的方式，就是回到前面，將 “SQL Server 驗證” 改成 “Windows 驗證” 。



改成 Windows 驗證之後，資料表就可以展開了，如下圖：



建立好這個連結之後，稍後我們就可以用選取的方式編寫程式。

在開始之前，我們先說明 ADO.NET 的架構

Visual Studio 是透過 ADO.NET來完成資料庫的存取，ADO.NET分成 .NET Framework Data Provider以及 DataSet兩個部分。其中 .NET Framework Data Provider 是存取料來源的一組類別程式庫，包含了 Connection、Command、DataAdapter、DataReader 物件。另外 DataSet 是記憶體中的資料庫，可以離線來進行資料操作。

當我們想要取出資料庫中資料表的時候，可以透過 DataAdapter 物件中的 SelectCommand、InsertCommand、DeleteCommand、UpdateCommand四個命令(後面三個就是我們每次在 Thingworx 裡面做的三個基本按鈕，這裡都已經內建了)，將資料表填入記憶體中的 DataSet (有點像是 InfoTable 的作用)，然後我們就可以透過 GridView 的方式顯示出來。

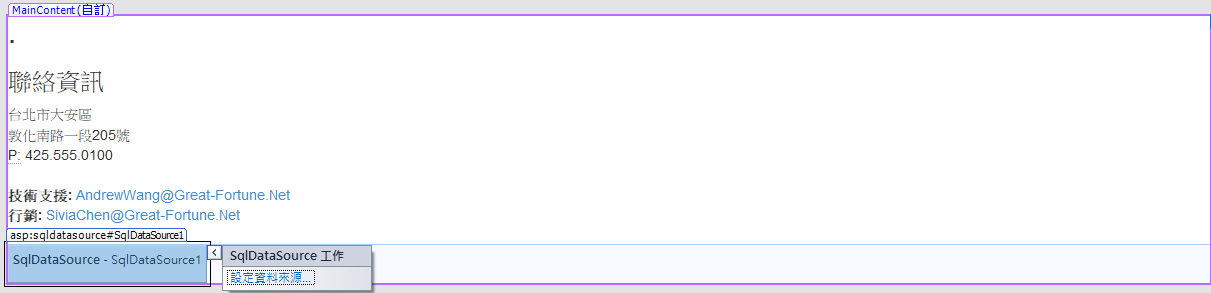
特別說明一下，DataAdapter 所提供的 SelectCommand、InsertCommand、DeleteCommand、UpdateCommand四個命令，相對於 Thingworx，是一模一樣的，只是 Thingworx 把 Select定義成 Query；Insert、Delete、Update才定義成命令。

接下來我們要用視覺化的方式來連資料庫，無須撰寫程式。首先看一下 WebForm 提供那些與資料庫相關的可視化物件：

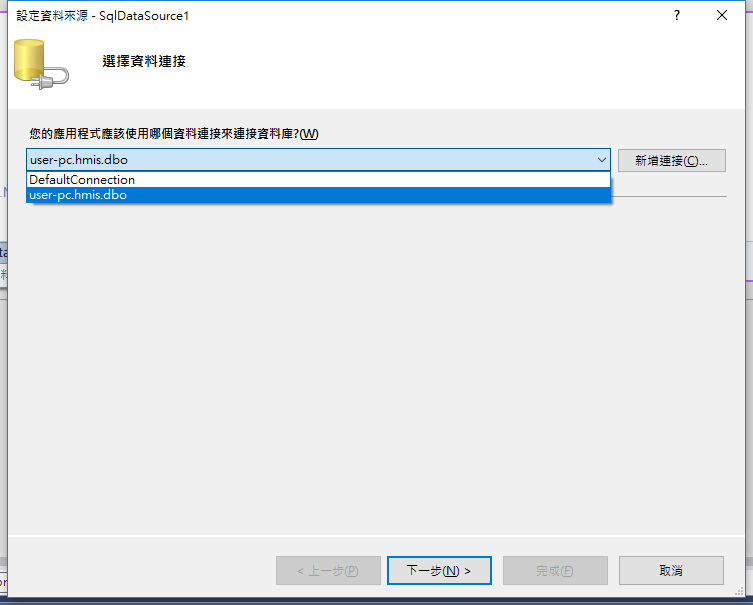


有經驗的程式設計師應該可以一眼看出來這些元件大概的用途，如果我們要在 Contact 頁面上面顯示資料表，顯示的元件可以用 GridView，而內容的來源應該就是 SqlDataSource。

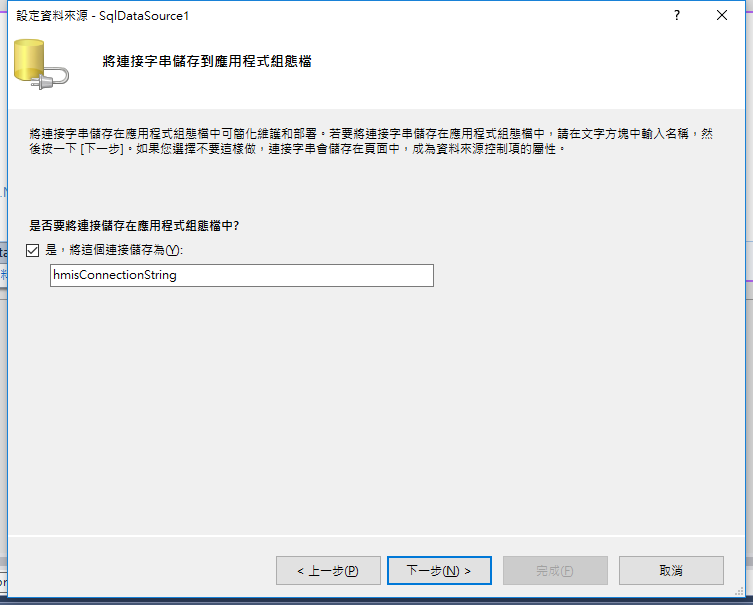
我們就用此邏輯來操作，先建立一個 SqlDataSource。我已經偷偷在 Contact.aspx的下方加了一個 <div>，方便我們加入元件。直接把 SqlDataSource 從工具箱拖曳到頁面的 <div> 空白處，就會出現下面的畫面：



按下設定資料來源，第一步，選擇我們在前面所設定的資料連結：



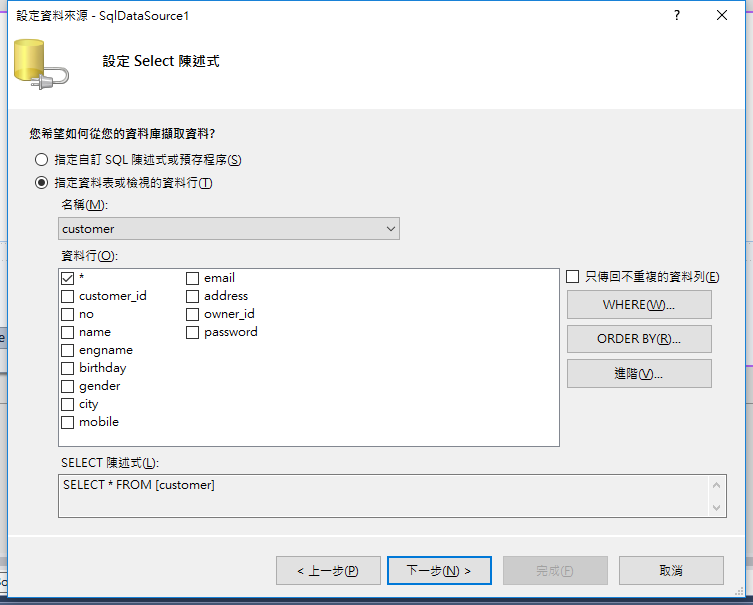
然後儲存在程式組態之中：



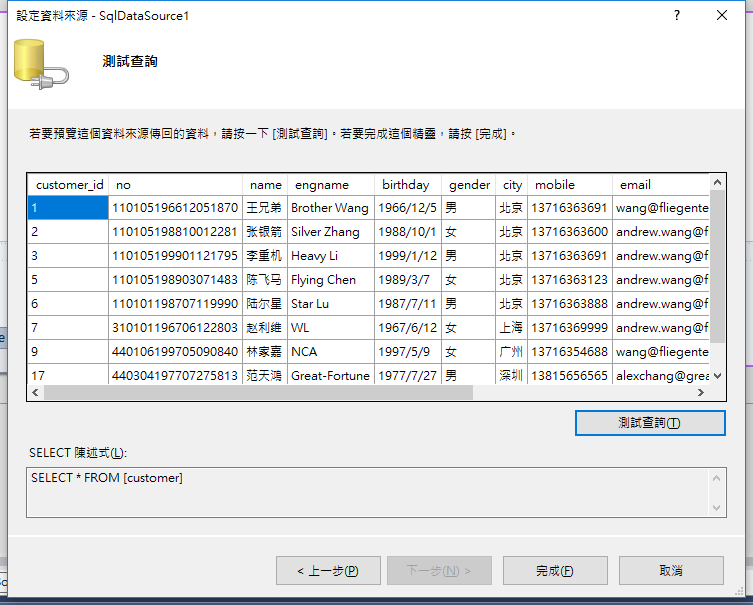
接下來就會出現設定 Select 陳述，我們選擇好資料表之後，就會自動生成下面的陳述：

SELECT \* FROM [customer]

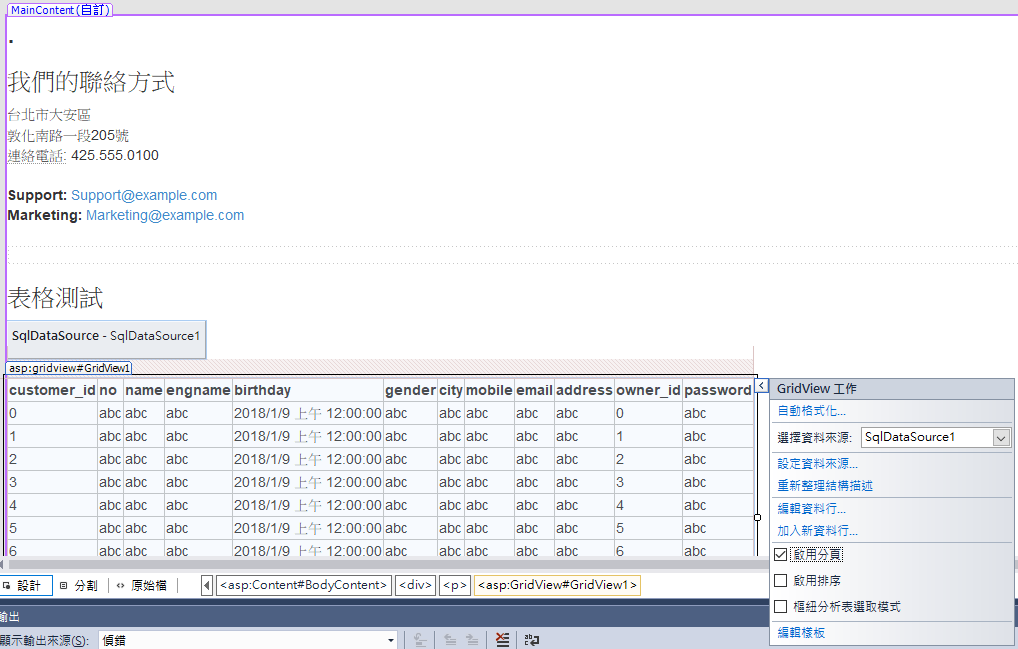
如下圖：



測試查詢無誤之後，按下完成：

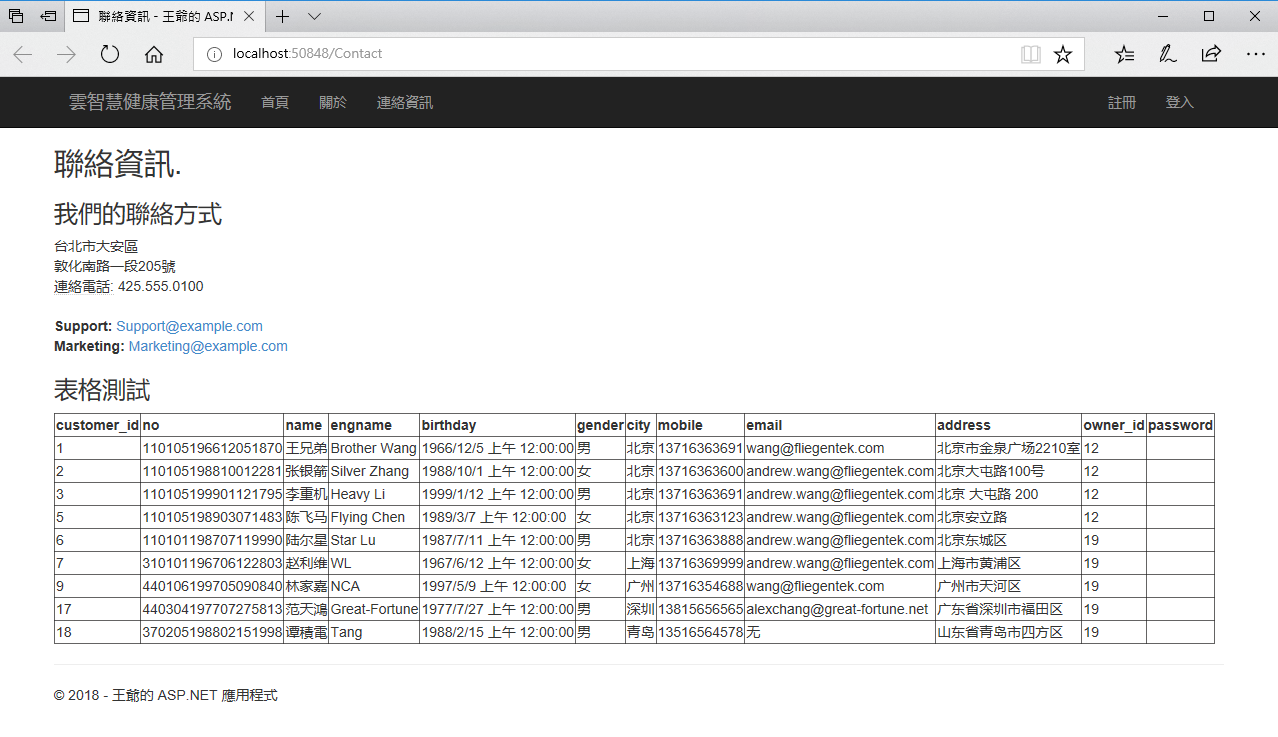


接下來我們直接拉 GridView 進來：



只要簡單的選取剛才設定的 SqlDataSource1，就完成了。

請參考我的執行結果：



#### 一之五、由資料庫建立動態網頁(WebForm)

很多時候，我們的網頁資料是來自於資料庫，例如健檢優惠組合或是問卷調查，這些資料的內容原本是在資料庫裏面的一筆記錄，我們從資料庫把記錄讀取出來之後，再用 html 的方式呈現。

在說明動態網頁之前，我們先從靜態網頁開始，我會按照 VS 的操作方式，讓大家了解如何正確地建立 .aspx 網頁。

我們還是使用 AndrewHmisFormLogin 這個方案為範例來說明。當我們打開方案之後，有兩個檔案我們之前有介紹，那就是 Default.aspx 與 Site.Master，先來看一下 Default.aspx 程式碼：

<%@ Page Title="首頁" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<div class="jumbotron">

<h1>云智慧健康管理系统</h1>

<p class="lead">云智慧体检全国各直营分院2018年元旦期间特惠</p>

<p><a href="http://www.asp.net" class="btn btn-primary btn-lg">详细内容 &raquo;</a></p>

</div>

<div class="row">

<div class="col-md-4">

<h2>健康体检</h2>

<p>

“健康体检”理念的开创者，提供个性化体检服务方案、健康评估、沙龙式会员管理、科学化疾病预警、健康干预、健康处方、便捷化专家门诊等服务。

</p>

<p>

<a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301948">详细内容 &raquo;</a>

</p>

</div>

<div class="col-md-4">

<h2>迈向高端</h2>

<p>

向高端人群提供全面的深度体检、私人医生、个性定制健康管理方案、健康养护、绿色就医通道等全维度健康调养。

</p>

<p>

<a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301949">详细内容 &raquo;</a>

</p>

</div>

<div class="col-md-4">

<h2>深度体检</h2>

<p>

开设“深度体检、私人医生服务、健康养护、全周期综合抗衰老、国内外就医转诊”五位一体健康管理体系。并添设“慢病管理、辅助生殖、生物治疗”三大核心功能，为客户提供更多国际领先的新药品、新器械、新技术。

</p>

<p>

<a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301950">详细内容 &raquo;</a>

</p>

</div>

</div>

</asp:Content>

這段程式碼裡面，有標準的 Template 可以讓我們在其他網頁使用，除此之外，裡面也用到了 BootStrap 自適應的寫作技巧，執行之後把瀏覽器放大縮小就知道了：

<div class="row">

<div class="col-md-4">

…

</div>

</div>

要了解更多 BootStrap 的用法，請自行上網找資料，我們要談的是另一個重點，就是 Site.Master，它的用處就如同 Thingworx 的 Master Mashup，你所設計的任何網頁，如果希望成為 Site.Master 的子畫面，就必須在第一行加入 MasterPageFile="~/Site.Master"，在 Default.aspx 的第一行就有。

如果我希望在 Site.Master 裡面，加入一個 “健檢套餐” 以及 “問卷調查” 的選單，只要稍微修改成下面的程式碼即可：

<ul class="nav navbar-nav">

<li><a runat="server" href="~/">首页</a></li>

<li><a runat="server" href="~/Combo">健检套餐</a></li>

<li><a runat="server" href="~/Survey">問卷調查</a></li>

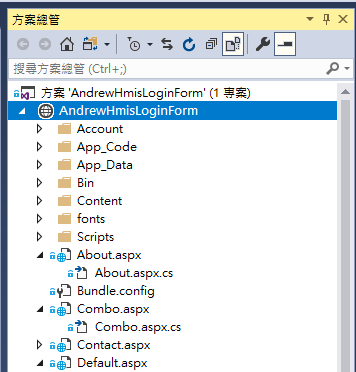
<li><a runat="server" href="~/About">关于</a></li>

<li><a runat="server" href="~/Contact">联络我们</a></li>

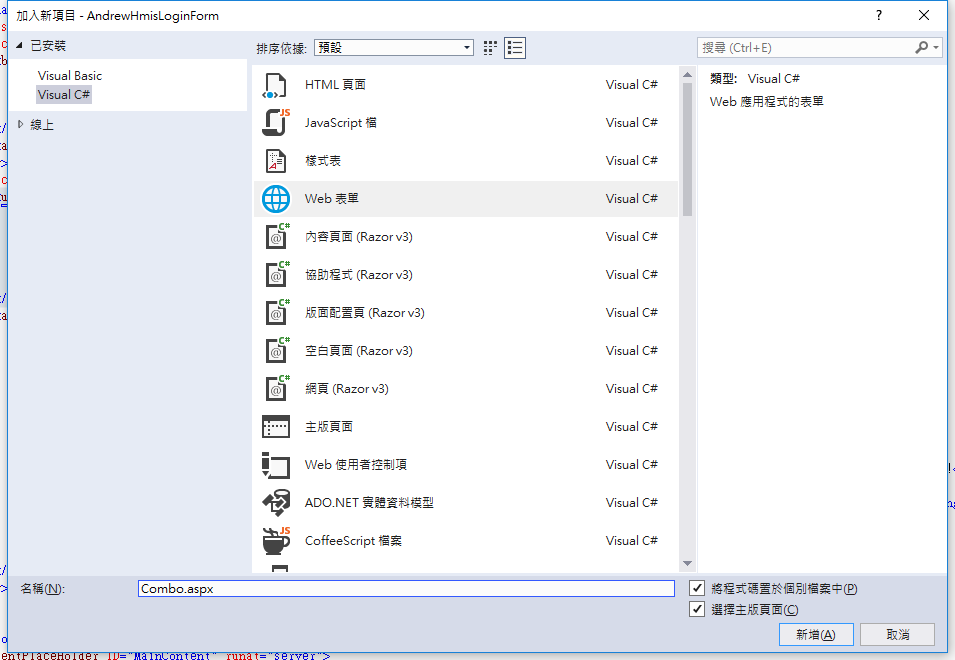
</ul>

可以看到，原本只有首頁、關於、和聯絡我們，現在已經加入了健檢套餐與問卷調查，當然，檔案名稱要事先輸入，然後接下來，我們就來按照前面取的方案名稱建立新的網頁。

接下來我們要建立網頁，網頁放在 Default.aspx 同一層，所以請選取 AndrewHmisLoginForm 專案(請參考下圖，要點在有地球圖示的那一個才是專案，最上面的是方案，兩者不同)，按滑鼠右鍵，選擇加入／加入新項目：



(注意是選第二行)



加入新項目時，必須選取 Web 表單，輸入檔案名稱，最重要的是 “選擇主版頁面” 一定要打勾，然後選擇 Site.Master，這樣我們新建的頁面才會跟 Site.Master 連結起來。

Combo.aspx 建立好之後，我們先按照 Default.aspx 的寫法，製作一個靜態網頁，以下是程式碼，可以跟 Defaul.aspx 比較一下便知異同：

<%@ Page Title="健检套餐" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeFile="Combo.aspx.cs" Inherits="Combo" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<div class="row">

<div class="col-md-4">

<br />

<asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>

</div>

<div class="col-md-8">

<h2>职场高端体检套餐</h2>

<p>职场高端体检套餐简介职场高端体检套餐简介</p>

<h3>标准价：￥4288元　点击进入商城，价格有惊喜哦！</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<br />

<asp:Calendar ID="Calendar2" runat="server"></asp:Calendar>

</div>

<div class="col-md-8">

<h2>感恩父母高端体检套餐</h2>

<p>检项表项目名称项目内容项目意义通用项目一般情况身高、体重、血压、体重指数体重指数是反映全身肥胖程度，更精确的测量指标。内科检查心率、心律、心脏杂音、心界、肺脏、肝脏、脾脏、肾脏、肠鸣音、营养发育、神经系统、生理...</p>

<h3>标准价：￥3858元　　点击进入商城，价格有惊喜哦！</h3>

</div>

</div>

</asp:Content>

執行結果如下(左邊我只是暫時用兩個月曆物件，可以改成圖片檔)：



靜態網頁設計容易，但是有一個問題，就是當我們在資料庫裏面的套餐有所修改的時候，資料庫變動了，網頁則需要自己手動重新編寫，這樣其實很麻煩。如果我們把網頁變成動態的設計模式，也就是說，網頁裡面原本沒有東西，當網頁執行的時候，才開啟資料庫，從資料庫中把套餐內容讀出來，然後再升成網頁，顯示在客戶端，這樣豈不是很酷。

所以我們就接續這個網頁的下面，呼叫一個 C# 程式，讓這個程式幫我們產生網頁內容。請將原本的程式碼加一行：<% =MyString() %>

<%@ Page Title="健检套餐" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeFile="Combo.aspx.cs" Inherits="Combo" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<div class="row">

<div class="col-md-4">

<br />

<asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>

</div>

<div class="col-md-8">

<h2>职场高端体检套餐</h2>

<p>职场高端体检套餐简介职场高端体检套餐简介</p>

<h3>标准价：￥4288元　点击进入商城，价格有惊喜哦！</h3>

</div>

</div>

<div class="row">

<div class="col-md-4">

<br />

<asp:Calendar ID="Calendar2" runat="server"></asp:Calendar>

</div>

<div class="col-md-8">

<h2>感恩父母高端体检套餐</h2>

<p>检项表项目名称项目内容项目意义通用项目一般情况身高、体重、血压、体重指数体重指数是反映全身肥胖程度，更精确的测量指标。内科检查心率、心律、心脏杂音、心界、肺脏、肝脏、脾脏、肾脏、肠鸣音、营养发育、神经系统、生理...</p>

<h3>标准价：￥3858元　　点击进入商城，价格有惊喜哦！</h3>

</div>

</div>

<% =MyString() %>

</asp:Content>

上面是原本的靜態網頁，而倒數第二行就是我們的動態網頁了。意思是，呼叫 MyString() 這個方法。請打開 Combo.aspx.cs，將下面的程式碼加在 Page\_Load() 的後面：

protected string MyString()

{

string cn = "Data Source=USER-PC;Initial Catalog=hmis;Integrated Security=True";

SqlConnection db = new SqlConnection(cn);

db.Open();

SqlCommand cmd = new SqlCommand("select combo\_no, name, gender, price, start\_date, end\_date, information from combo", db);

SqlDataReader dr = cmd.ExecuteReader();

string data = "<table border=3><tr>\n";

for(int i = 0;i < dr.FieldCount; i++)

{

data += "<td>" + dr.GetName(i) + "</td>";

}

data += "</tr>\n";

while (dr.Read())

{

data += "<tr>";

for(int i = 0;i < dr.FieldCount; i++)

{

data += "<td>" + dr[i] + "</td>";

}

data += "</tr>";

}

data += "</table>\n";

string myText;

myText = "<h1>健檢套餐在資料庫中的內容</h1>\n";

myText += "<form>\n";

myText += "<p>" + db.Database + "资料库已开启</p>\n";

myText += "<p>本程式为测试程式，开启资料库之后，透过 Command 物件以及 DataReader 物件独取资料</p>\n";

myText += "<p>资料库的栏位有：</p>" + data;

myText += "<br />\n<input type = \"submit\">\n";

myText += "</input>\n";

myText += "</form>\n";

db.Close();

return myText;

}

並且在前面的宣告中加上這兩行：

using System.Data;

using System.Data.SqlClient;

先看一下執行的結果：



上半部是原本的靜態內容，下半部就是從資料庫中擷取的內容。當然，我只是用 table element 來展示，沒有做任何美工，如果要讓這些內容顯示的跟上面的靜態內容一模一樣，當然也沒有問題。要讀懂上面的程式碼，需要對資料庫的操作有所認識，接下來我會詳細說明。

##### 引用 ADO.NET 命名空間

當我們使用到 ADO.NET 所提供的功能的時候，第一步就是要引用正確的命名空間，例如：

using System.Data;

using System.Data.SqlClient;

System.Data 是 ADO.NET 的核心包含 ADO.NET 架構的基礎類別、也包含了存取資料的主要方法。

接下來就是看你所使用的資料庫連接方式，如果是使用 OLE DB的方式連接資料庫，就引用 System.Data.OleDb，如果是使用 SQL Client 的方式，就引用 System.Data.SqlClient，這兩種方法是二選一的，所以在下面的範例中，我會以 SQL Client 的方式為主，但是會順帶說明另一種方式的寫法。

##### Connection 物件

連結資料庫有四個步驟，分別由四種物件來達成，分別是 Connection、Command、DataReader、DataAdapter。我們可以這樣來記，先連接資料庫、然後下 Select 命令、然後把資料讀出來、或針對DataSet 對資料表做更新。

如果是使用 OLE DB 的方式，對應的三物件為：OleDbConnection、OleDbCommand、OleDbDataReader、OleDbDataAdapter；如果是用 SQL Client 的方式，則為：SqlConnection、SqlCommand、SqlDataReader、SqlDataAdapter。

標準寫法如下(我們先談到 DataReader)：

string cn = "Data Source=USER-PC;Initial Catalog=hmis;Integrated Security=True";

SqlConnection db = new SqlConnection(cn);

db.Open();

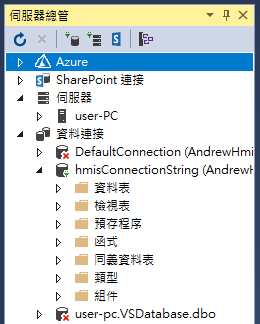
SqlCommand cmd = new SqlCommand("select combo\_no, name, gender, price, start\_date, end\_date, information from combo", db);

SqlDataReader dr = cmd.ExecuteReader();

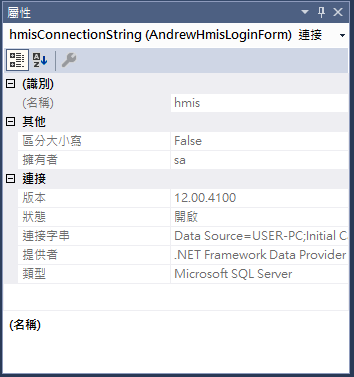
// 程式碼寫在這

db.Close();

這個程式框架式固定的，要開啟任何資料庫，都是這樣寫，處理完畢之後，記得 Close 資料庫。唯一要知道的，就是 Connection 的字串，透過這個字串才能開啟正確的資料庫。至於這個字串如何產生？不用自己寫，我們利用伺服器總管中的資料連接來產生。



我們透過VS提供的工具，來連接資料庫，請參考我的一之四的分享，連接成功之後，點選連好的資料庫，例如：hmisConnectionString，然後參考屬性裡面的連接字串，如下圖：



你就可以把字串複製下來，貼到程式碼中了。千萬不要自己寫。

##### Command 物件

Command 物件只有一個重點，就是 SQL Statement，SQL Server 的 SQL語法跟 MySQL 的語法會有一些小差異，這個部分我們就不詳加說明，請自行參考 SQL 語法的書籍。

SQL Command 有兩大類，一是資料庫的查詢(Query)：Select；另一類則是資料庫的操作：Insert、Update、Delete。當我們用 Select 做資料查詢的時候，我們會使用 cmd.ExecuteReader() 來執行，如果是對資料庫做操作的時候，則會使用 cmd.ExecuteNonQuery() 來執行。關於這點，我們回想一下 Thingworx 中也有 SQL Query 與 SQL Command 的區分，應該不難了解其中的差異。

程式的寫法當然也有所不同，cmd.ExecuteReader() 才會有傳回值：

SqlDataReader dr = cmd.ExecuteReader(); //需要 DataReader

cmd.ExecuteNonQuery (); //不需要 DataReader

##### DataReader 物件

Select 出來的資料，我們想像成一個表格(Grid)，會有若干筆記錄(Records)，而 DataReader 物件可以讓我們逐筆讀取資料記錄：

SqlDataReader dr = cmd.ExecuteReader();

接下來的重點，就是如何讀取每一筆記錄，其實只要簡單的透過一個 while() 迴圈即可：

while (dr.Read())

{

data += "<tr>";

for (int i = 0; i < dr.FieldCount; i++)

{

data += "<td>" + dr[i] + "</td>";

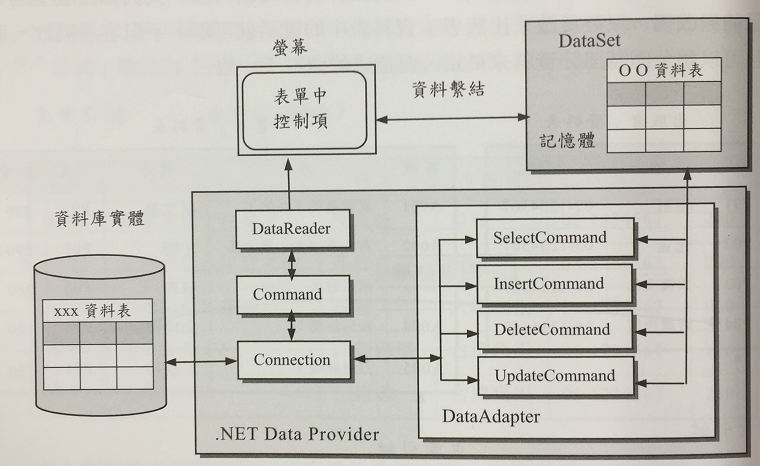
}

data += "</tr>";

}

其中 dr[i] 是讀取第 I 個欄位的資料 (while 裡面的 for 迴圈，就是抓出每個欄位)，也可用 dr[“欄位名稱”] 來讀取某個欄位的資料(就不需要 for 迴圈)。

DataReader 是逐筆讀取資料，請特別注意逐筆的意義，就是每次只從資料庫裏面讀出一筆資料存放於記憶體之中，如果我們想一次將整個查詢結果，就必須使用 DataAdapter 與 DataSet 物件了，如下圖：



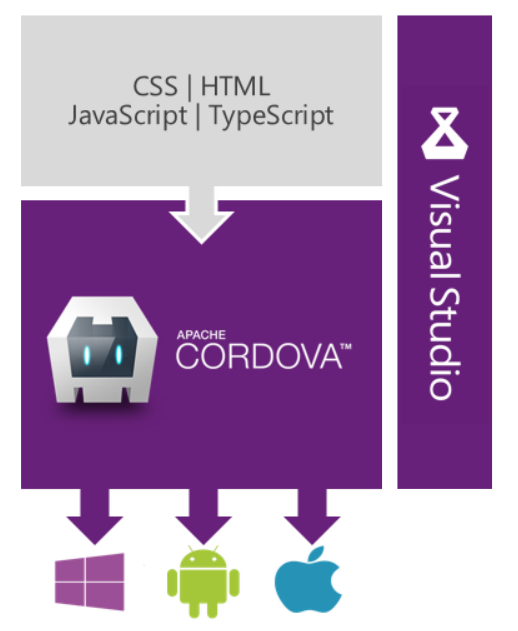
目前為止，我們只用到資料庫的簡單操作，所以走的是左半部的路徑 (Connection, Command, DataReader) ，當我們要搭配 WebForm 的 DataGridView 這種控制項的時候，才會走右半部的路徑。其實我們在上一篇一之四的文章中，已經說明過了，也就是說我們可以不需要編寫程式，也可以利用 DataGridView、DataSet、DataAdapter 來顯示資料。當然，我們也可以用寫程式的方式達成。

### 二、第一個 Cordova APP

現在最流行的話題就是跨行動裝置 App 開發: 使用 Visual Studio 2017 + Apache Cordova (PhoneGap)。我就不必單獨介紹 Cordova，相信很多書籍都有談到，我的重點會在於 Visual Studio + Cordova。

回顧一下歷史，2013 年 11 月時，微軟宣佈了正式與 Xamarin 合作，在 Visual Studio 上可以使用 C# 作為程式語言來開發跨 Windows、Android 及 iOS 行動平台的 app，所以，在 Visual Studio 2017 裡面我們可以選到 Xamarin的模塊。它是基於 MVC 架構的 C# 模塊。

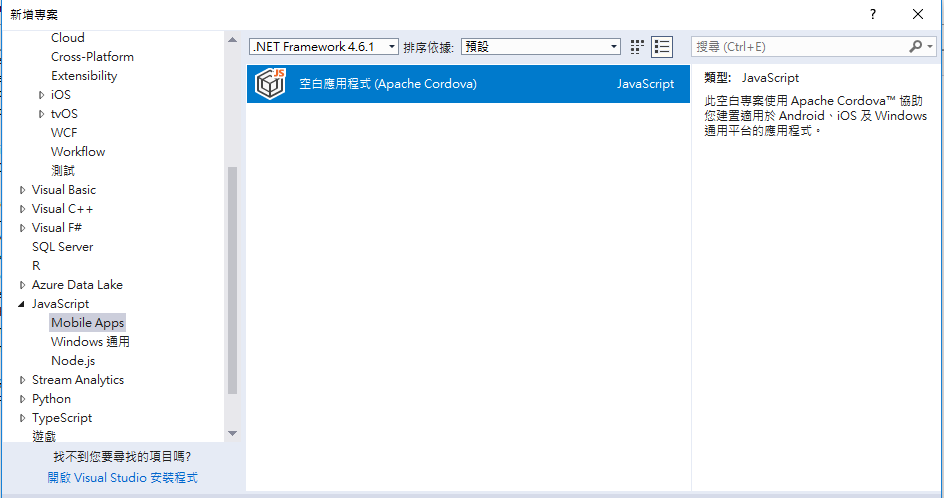
而與 Apache Cordova 的合作則是另一個跨行動平台開發的方向（使用 JavaScript/HTML5）。在 TechEd North America 2014 研討會期間，Microsoft Open Technology 宣佈了 Visual Studio 與 Apache Cordova 合作的計劃，而 Apache Cordova 則是大家熟知的 PhoneGap 目前的專案名稱，與 Visual Studio 合作的意義便是 －－ 在 Visual Studio 裡可以使用 JavaScript/HTML5 技術來開發橫跨 Windows Phone、iOS 及 Android 的 app。



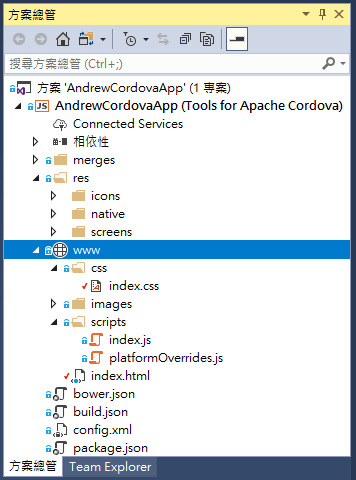
當然，我們最後的決定是使用 JavaScript/HTML5 來開發，因為這樣的架構真的比較簡單，可以讓我們更快速的看到成果。

心動不如行動，我們現在馬上來寫第一個 Cordova APP，基於 VS 的整合開發平台。

新增一個方案，選擇 “空白應用程式(Apache Cordova)” ，如下圖：

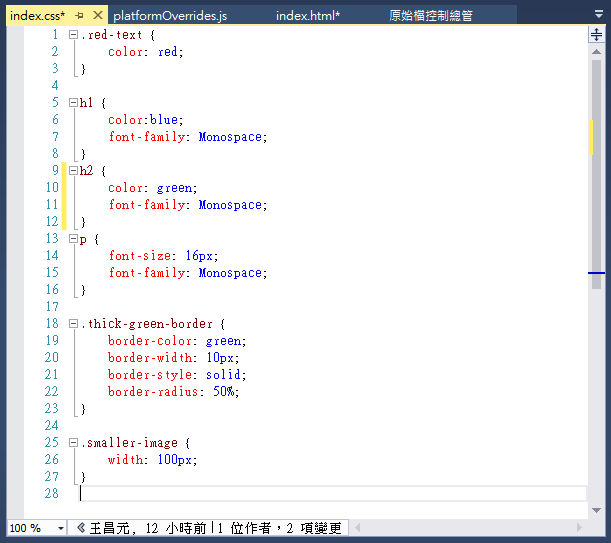


方案建立完畢之後，會得到這樣的一個基本架構：

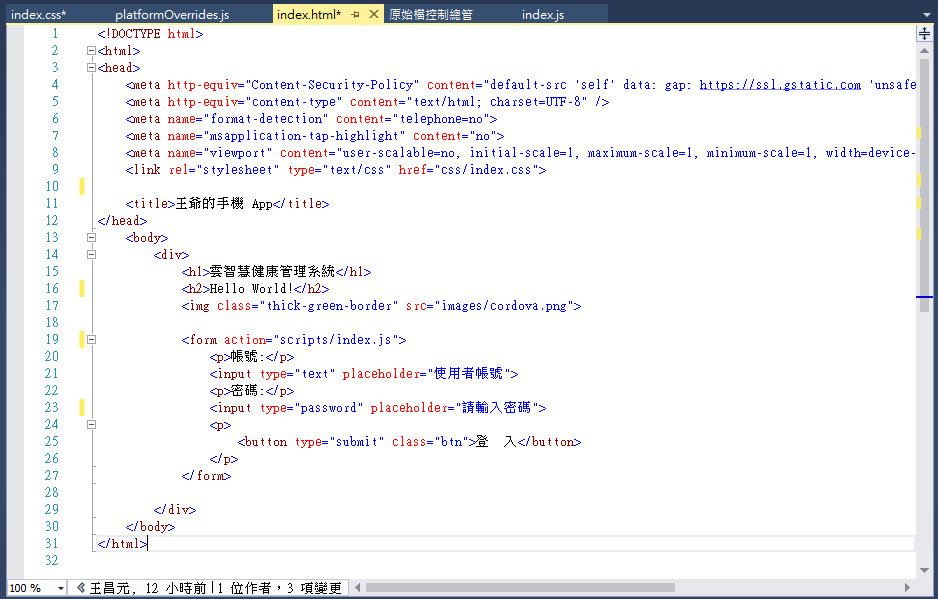


這個架構相當簡單，其實只有 index 一個網頁而已，叫做 index.html，與它相關的檔案就是 index.css 與 index.js，三個檔案分別存放我們撰寫的 html5、CSS3 與 JavaScript。

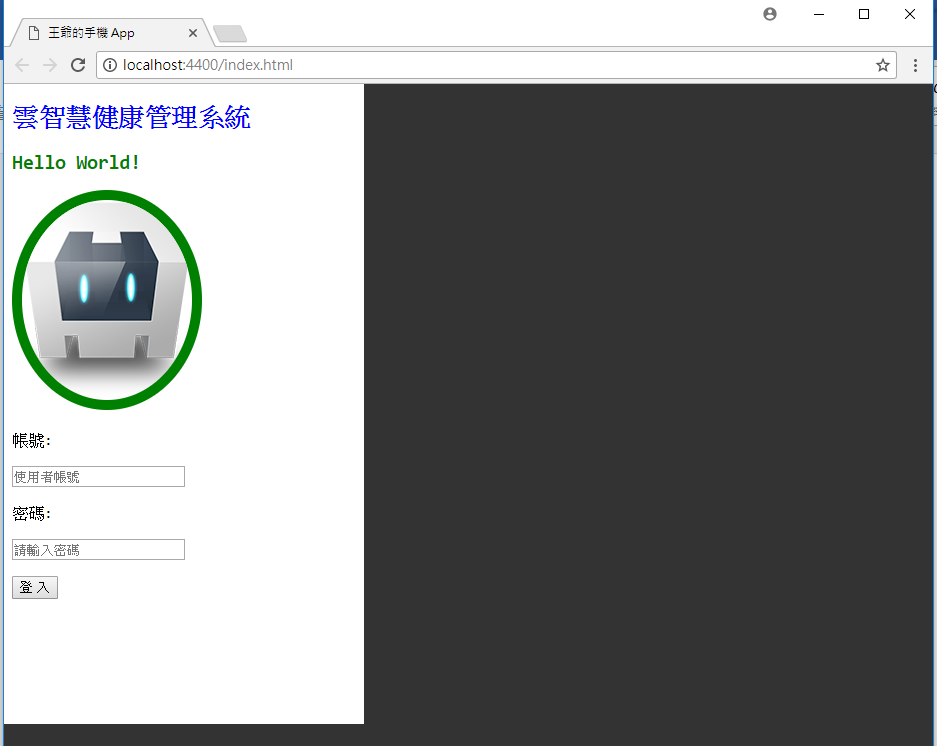
VS 自行產生的程式碼很複雜，基本上對我們的 Hello World 程式沒有用處，讓我們從頭來。先打開 index.css，全部刪除重寫，我只放了一些最基本的CSS定義，如下圖：



接著編輯 index.html，裡面包含了所有鏈結到 index.css 與 index.js 的片段，如下圖：

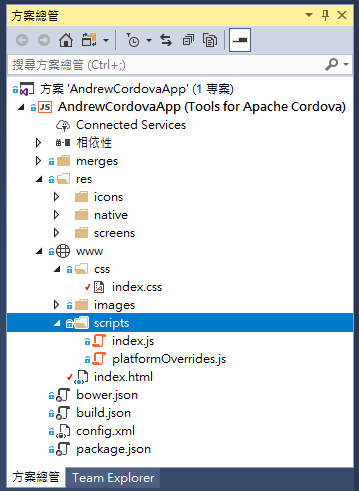


直接按下執行：

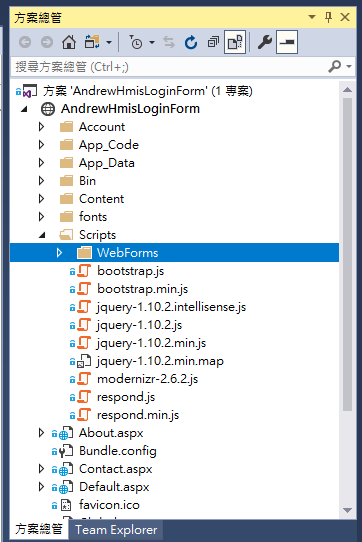


#### 二之一、jQuery 函數庫的安裝

其實有一件事情我個人覺得很奇怪，在我們剛才建立出來的 Apache Cordova 方案裡面的 Scripts 目錄中，其實只有兩個檔案，其中一個是 index.js，並沒有加入一些 jQuery 的函數庫。反觀當我們用 WebForm 建立方案之後，Scripts 目錄裡面則是滿滿的函數庫：



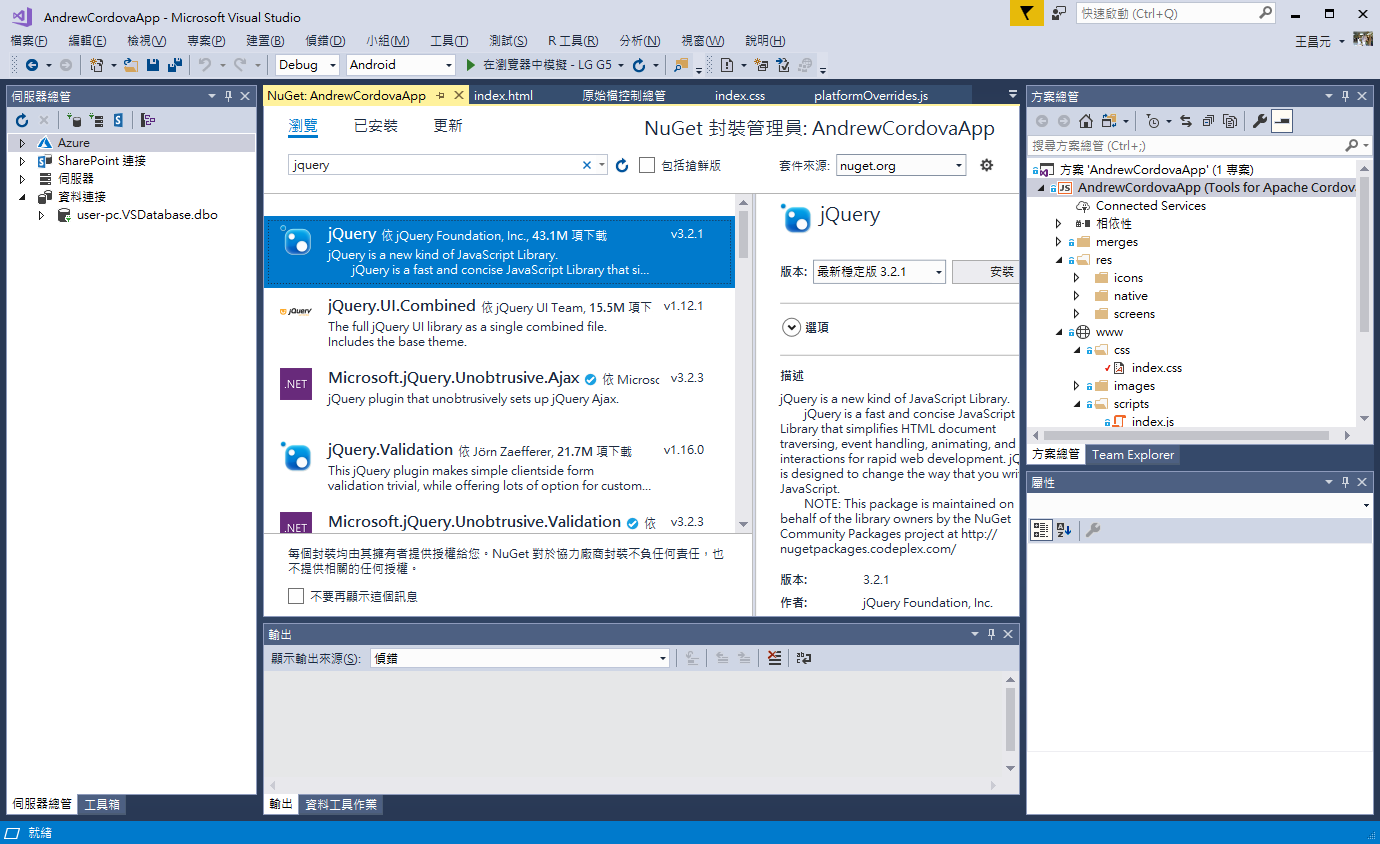
(Cordova)



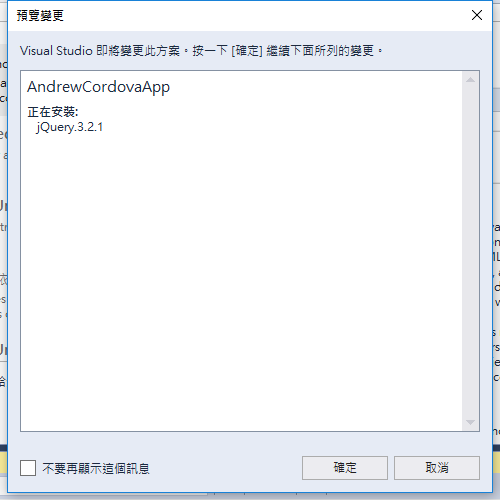
(WebForm)

由於我們要寫 jQuery 的程式，既然沒有，我們就自己加入吧。我們就用正統的 Visual Studio 的方式來加入。

在方案總管中選取方案按下滑鼠右鍵，選取NuGet 封裝管理員，出現下圖：

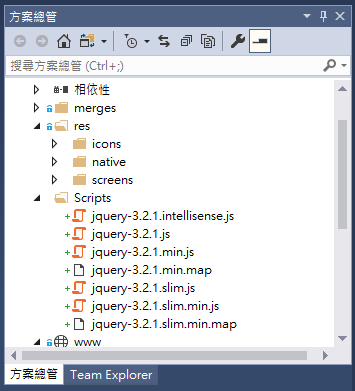


在搜尋裡面輸入 jQuery，會找到我們所需要的函數庫，先選取 jQuery ，然後按下右邊的[安裝]按鈕，就會出現下面的對話框：

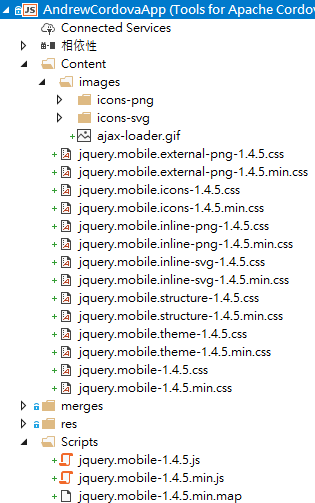


按下[確定]即可。安裝完成之後，會新增加一個 Scripts 的目錄，並多出

七個函數庫，如下圖：



同樣的方式，再找到 jquery.mobile，按下安裝。這裡所需要的時間比較長，因為會安裝的東西很多，除了 js 檔案之外，還有 CSS跟圖標檔案等等。請參考下圖：



完成此步驟之後，恭喜你，你就可以開始寫 jQuery Mobile 的程式了。

#### 二之二、連結jQuery函數庫

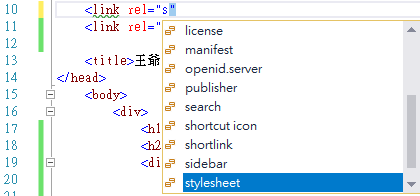
安裝好jQuery函數庫之後，要如何使用它呢？我們必須在程式裡面連結。開發基本的 jQuery Mobile 有三樣東西一定要連結，其中一個是 CSS 而另外兩個是 JS，這兩類東西的連接方式是不一樣的。

連接CSS 必須使用 link element 來宣告，在 index.html 的 head 裡面輸入以下的程式，在這裡我使用VS整合開發環境中的智慧選字功能來做，這個功能十分強大，所以在編程的時候切記要使用VS的編輯器來編輯，不要用其他第三方軟體。

當我們打入 <li 的時候，智慧選字系統已經啟動了，入下圖：



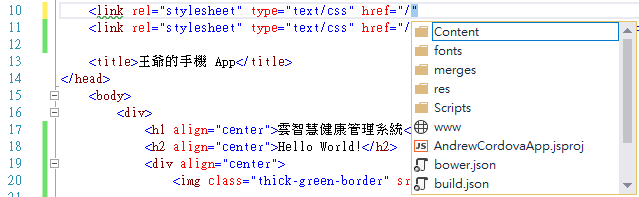
因此我們不必把字打完，直接按空白鍵，系統自動幫我們輸入，接著往下打 re，系統會自動選出 rel，同樣的，我們就直接輸入 = 就好，這時系統又會自動產生雙引號，並且列出所有的選項讓我們選擇：



按照這種方式，到最後要選擇檔案的時候，智慧選字就會列出目錄，讓我們選：



如果檔案在其他平行的目錄中，我們只要輸入 / 符號，則會自動跳到這個專案的根目錄：



要連結的CSS在Content 目錄之中，這樣我們就可以很輕鬆地找到：



選到檔案之後輸入 >，則自動完成：



說明一下：第一行是我們剛剛做的結果，第二行則是因為未來會用到自適應的功能，所以我就直接把 bootstrap 函數庫安裝之後也連結進來。

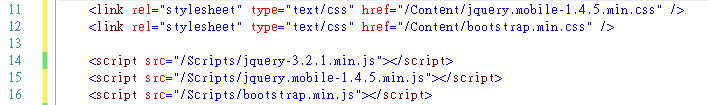
CSS 連結完成之後，接下來我們把兩個 script 也連結進來，使用 link element，script 則是放在根目錄下面的 Scripts 目錄裡面：



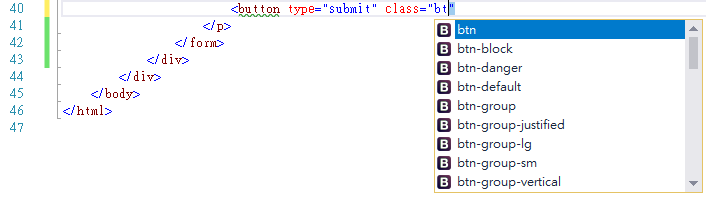
完成之後結果如下圖：



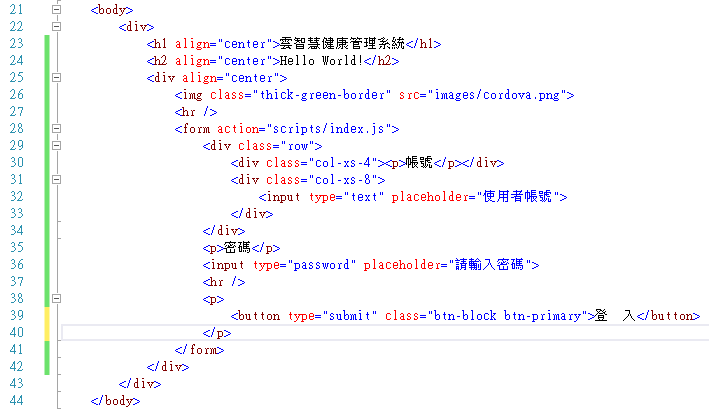
如果把 BootStrap 也算進來，就會是下面五個：



編成的時候也可以使用智慧選字功能，當我們匯入了額外的CSS的時候，選擇就會變多，例如我們要設計一個 button：



如果沒有連結 BootStrap，這裡是沒有 btn 選項的，匯入之後就可以選到。程式碼如下：



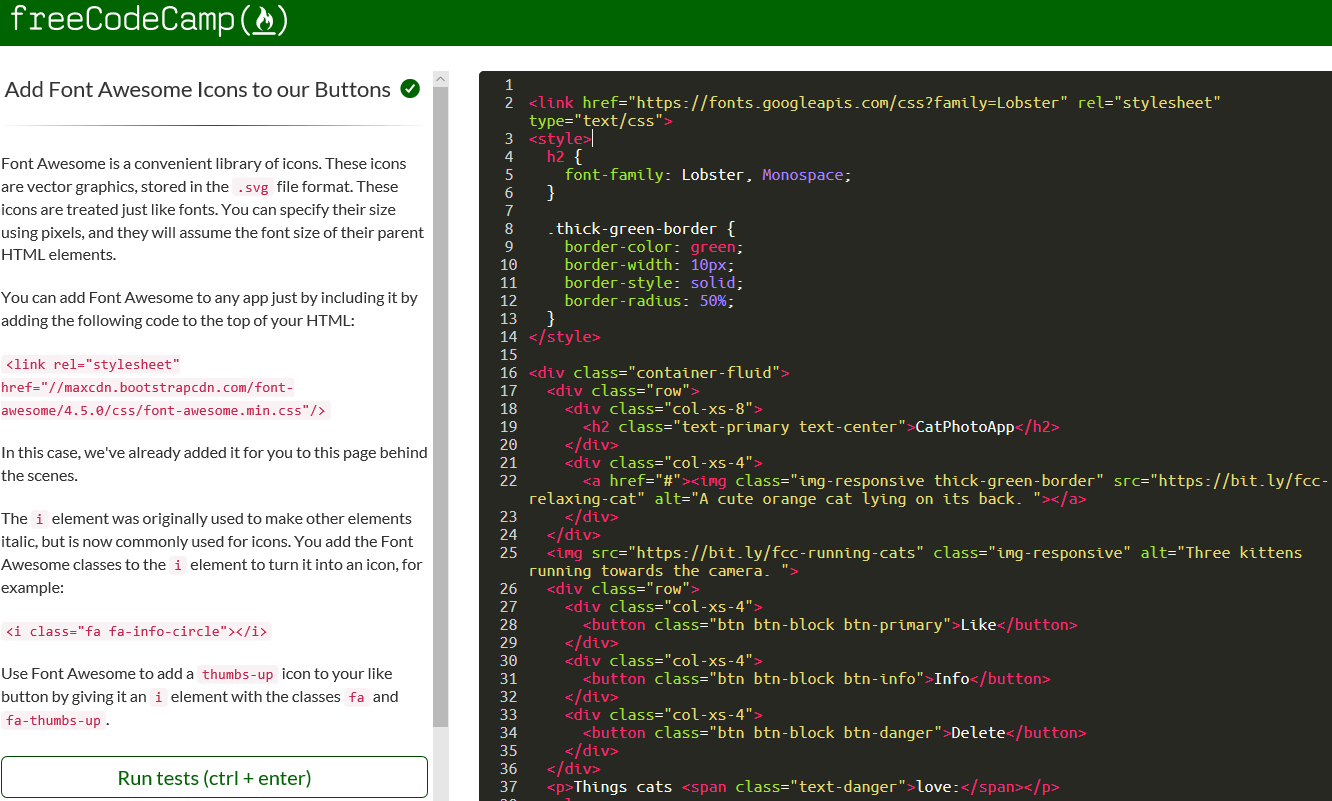
讓我們先看一下結果(測試)：



#### 二之三、安裝第三方CSS來加強 HTML5的功能

再上一個章節我們安裝了基本的 jQuery 的 JS 與 CSS，為了要做到自適應的頁面，又多裝了 BootStrap 套件，其實還有很多第三方的套件可以讓我們的網頁更豐富。

我在做 FreeCodeCamp 練習題的時候，看到一個主題如下：



Font Awesome 是疊加在 BootStrap 之上的一個 CSS，可以讓我們用最快的方式加入一些特別的圖示，當然，我們可以按照文章裡面說的加入一個 CDN 的連結，如下圖：

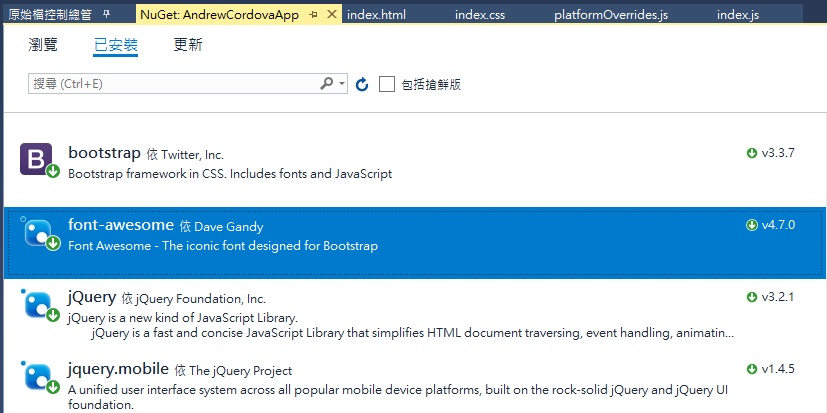


但是我還是希望我們可以用VS介面比較正統的方式來增加。同樣的，在專案名稱上面按滑鼠右鍵選 NuGet 管理員，搜尋 font awesome，我們就會找到這幾個：

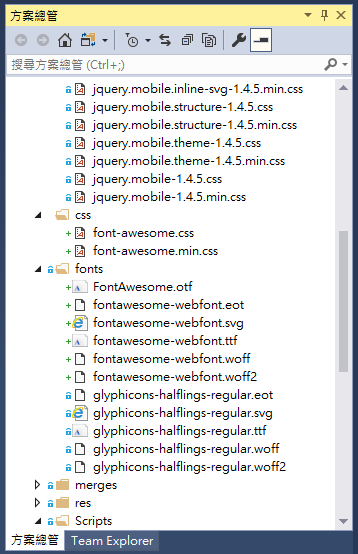


其中選取的這個 designed for BootStrap，看起來最像，先按下安裝。

安裝完成之後，我們可以在 “已安裝” 的畫面裡面看到我們從一開始到現在所有安裝的套件：



在看一下方案總管，Font Awesome 則是安裝在 css 這個目錄底下：

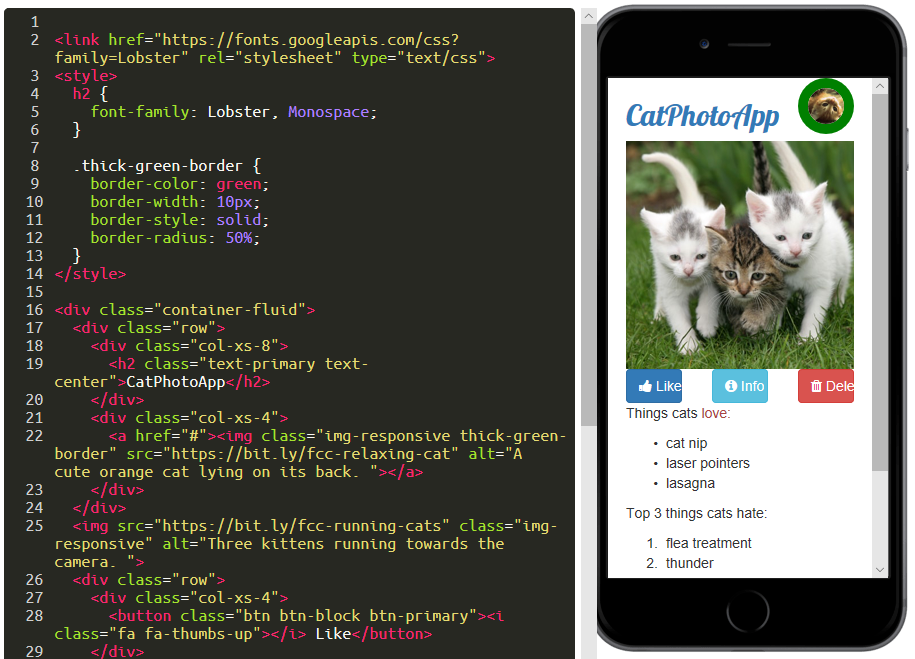


接下來回到 index.html，我們把連結加入：



(第三個)

接下來我們按照下圖 28 行的說明，將程式碼輸入：



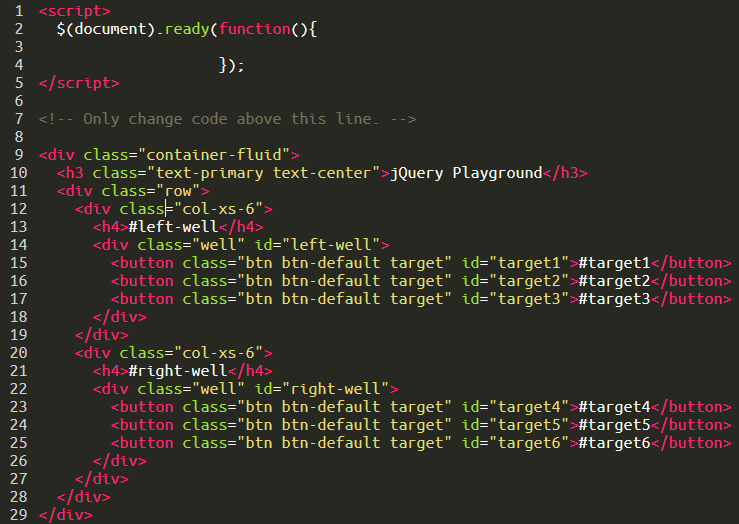
我們的程式碼如下：



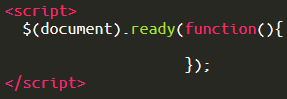
執行結果如下圖，是不是比較好看了些：



#### 二之四、jQuery 的基本架構



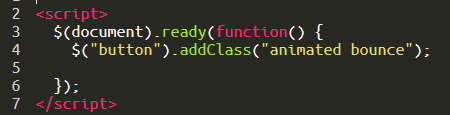
上圖是一個 jQuery 的基本架構，包在 script element 裡面，其他部分則是我們的 html，而在 script 裡面，就是 document ready function。如下圖：



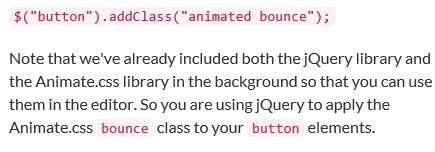
先不要去研究為何程式碼要這樣寫，我們先把英文翻譯成中文之後，就會瞭解其用意。Document ready function 就是網頁文件準備好之後的功能函數。其中出現的小括號與大括號，如果有程式設計的基礎，不難看出它是一種槽狀結構。

所謂 “網頁文件準備好之後的功能函數”，就是指當網頁完整的下載到本地端之後，才會開始執行的函數，因為如果沒有下載完全，執行會出錯誤。

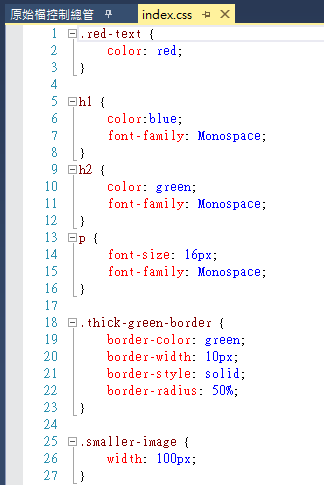
這只是個框架，我們的程式碼會寫在大括號之中。例如：



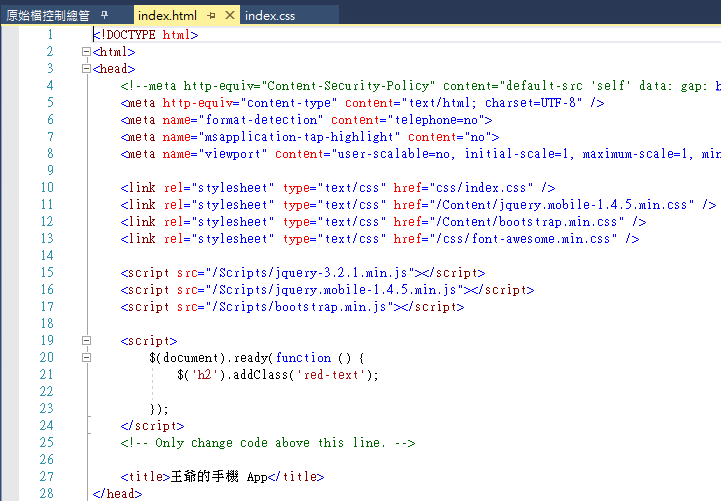
這個程式如果直接放到你的網頁裡面執行，會發現沒有任何反應，原因是 “animation bounce” 是 Animate.css 提供的額外功能，我們並沒有安裝，說明如下：



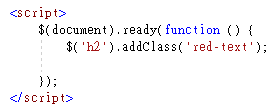
現在回到我的 AndrewCordovaApp 方案，首先，由於這個方案在建置的時候，會自動生成 index.html, index.css, index.js ，我建議先將index.css 清空重寫，如果你沒有重寫，則無法執行出你要的結果，所以我的index.css 裡面是自己會用到的東西，如下圖：



Index.html 則是將 body element 裡面的內容全部清空重寫，而 head 裡面的第一行，也就是程式碼的第四行，請刪除或是 Remark 起來，不然 jQuery 會無法執行，如下圖：



如同上面的程式碼，把自己的 link 與 script 連結進來之後，就可以撰寫自己的 jQuery 程式碼，它的位置也會在 head 裡面：



我們解釋過框架了，直接看框架裡面的程式。jQuery 的程式架構分成三個部分，分別是：

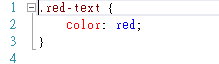
$(選擇器).動作(參數);

$(‘h2’).addClass(‘red-text’);

##### 選擇器

就是用來選取文件裡面的物件，先看一下下面的程式：

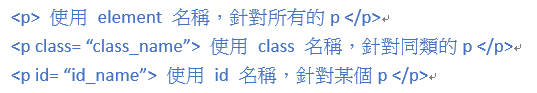
(index.css: 定義了 red-text 這個 class)



(index.html: 定義了 inline script，以及網頁內容)



我們在說明 CSS 的時候，有談過三個層次，分別是：



而jQuery 在選取物件的時候，會用下面的方式：

* 直接寫 element 名稱(都需要加上單引號或是雙引號)，如 h2，就是針對所有 h2 定義。
* 名稱前面加一個 “ . ” 就是 class 名稱，如 .red-text，就是針對所有 class= “red-text” 的 element定義。
* 名稱前面加一個 “ # ” 就是 id名稱，如 #cat-photo-form，就是針對唯一的 id= “cat-photo-form” 的 element定義。

這樣我們就可以很清楚的說明，$(‘h2’).addClass(‘red-text’); 之中的 $(‘h2’)，就是指在網頁中出現的所有的 h2，就是我們程式裡面的 Hello World!。

如果寫成 $(‘.myHeader’).addClass(‘red-text’);，則是選取所有 class=”myHeader” 的物件，就是程式碼中的 h2, h3。

若是 $(‘#myHello’).addClass(‘red-text’);，就是選擇 id=”myHello” 的物件。

##### 動作

$(‘h2’).addClass(‘red-text’); 中的 addClass 就是一個動作，這是 jQuery 函數庫裡面的一個方法，意思是把後面的參數加入 class 字串裡面，也就是說，原本 h2 裡面的 class="myHeader text-center" 會變成 class="myHeader text-center red-text"。

##### 參數

參數就不多解釋，就是上面所說的 red-text 字串。這個字串不是隨便亂寫的，而是在 CSS 裡面定義的，請看CSS內容，.red-text 是定義 color:red顏色用的，傳入這個字串目的，就是把 h2 的內容字串設定成紅色。

請看執行結果：



當然，如果將 $(‘h2’).addClass(‘red-text’); 改成 $(‘.myHeader’).addClass(‘red-text’); ，就會將所有 class=”myHeader” 的物件變成紅色。如下：



這麼一來，h2 h3 都變成紅色了。

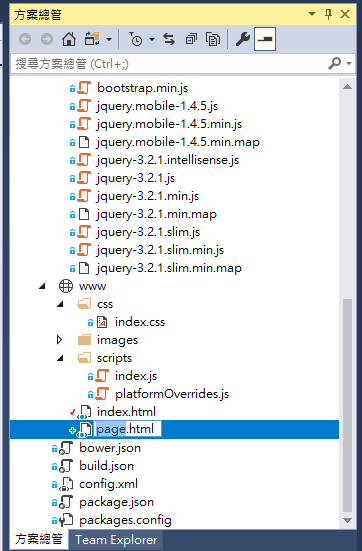
#### 二之五、新增FreeCodeCamp測試頁面

如果你們也有去做 FreeCodeCamp 裡面的練習，當你看到比較有趣的主題的時候，其實可以把該網頁納入你的 VS 測試專案之中，測試執行結果，這也算是一種實際的應用。

這裡的分享很重要，因為我會用VS整合開發平台的操作方式來做，對於初學VS的人，可以更快速的了解如何使用Visual Studio。

我們現在來增加一個新的 html 檔案，首先我們需要知道一般而言，html 是放在 www這個目錄底下的，也就是跟 index.html 同一個目錄。

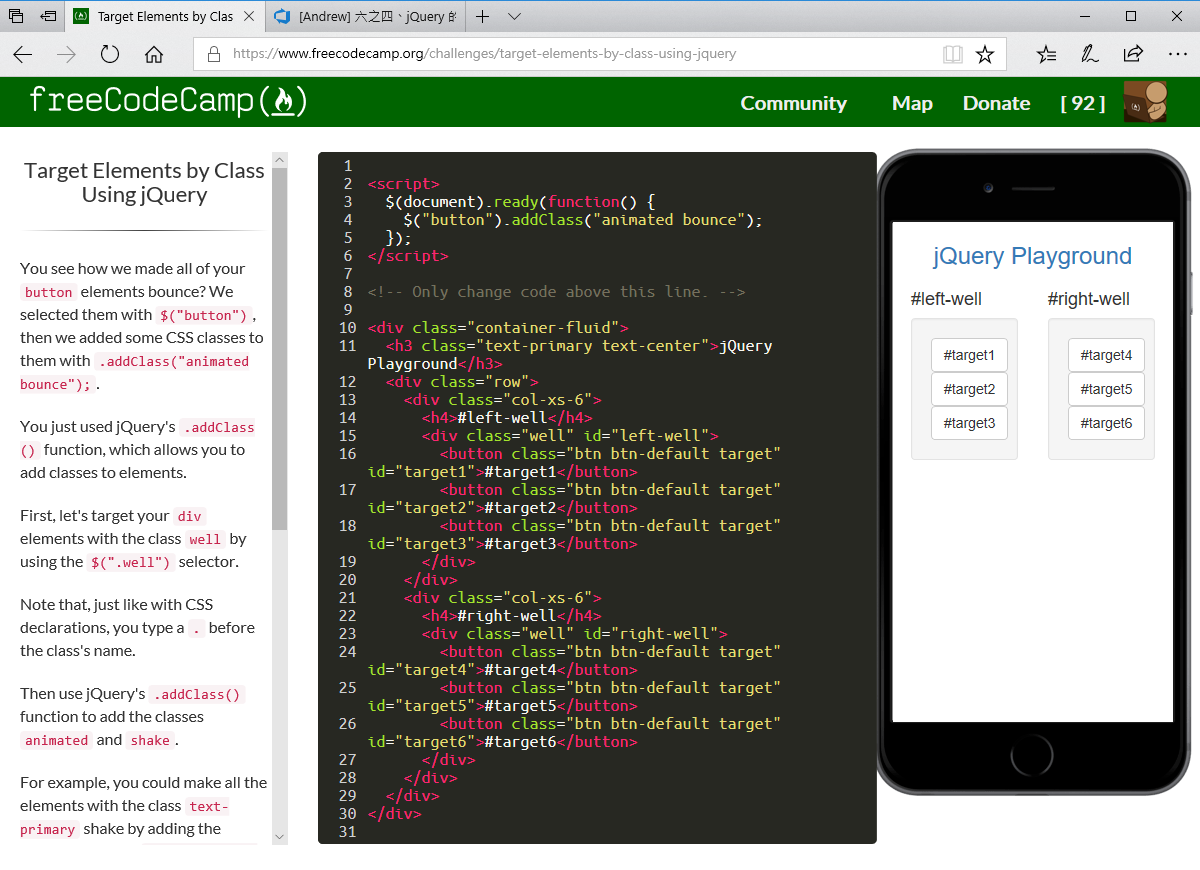
選取方案總管裡面的 www ，按下滑鼠右鍵選取 “新增項目”，選 html 檔案。這是會有一個 page.html 檔案產生，如下圖：



在 page.html 按下滑鼠右鍵選 “重新命名”，請將名稱改為 FreeCodeCamp。這是，在編輯區裡面就會產生一個空的 html，並包含了基本架構：



舉例，我們現在要做的練習是 Target Elements by Class Using jQuery ，如下圖：



練習裡面的程式碼如下：

<script>

$(document).ready(function() {

$("button").addClass("animated bounce");

});

</script>

<!-- Only change code above this line. -->

<div class="container-fluid">

<h3 class="text-primary text-center">jQuery Playground</h3>

<div class="row">

<div class="col-xs-6">

<h4>#left-well</h4>

<div class="well" id="left-well">

<button class="btn btn-default target" id="target1">#target1</button>

<button class="btn btn-default target" id="target2">#target2</button>

<button class="btn btn-default target" id="target3">#target3</button>

</div>

</div>

<div class="col-xs-6">

<h4>#right-well</h4>

<div class="well" id="right-well">

<button class="btn btn-default target" id="target4">#target4</button>

<button class="btn btn-default target" id="target5">#target5</button>

<button class="btn btn-default target" id="target6">#target6</button>

</div>

</div>

</div>

</div>

我們要將它複製到VS裡面了html檔案，要稍微做一些修改。首先，由於我們會用到 jQuery 程式庫，請先將 index.html 裡面的程式庫連結複製到 head 裡面，然後再將上面程式的 script 部分也複製到 head 裡面，結果如下：

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<meta name="format-detection" content="telephone=no">

<meta name="msapplication-tap-highlight" content="no">

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="stylesheet" type="text/css" href="/Content/jquery.mobile-1.4.5.min.css" />

<link rel="stylesheet" type="text/css" href="/Content/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/css/font-awesome.min.css" />

<script src="/Scripts/jquery-3.2.1.min.js"></script>

<script src="/Scripts/jquery.mobile-1.4.5.min.js"></script>

<script src="/Scripts/bootstrap.min.js"></script>

<script>

$(document).ready(function () {

$("button").addClass("animated bounce");

});

</script>

<title>FreeCodeCamp 測試網頁</title>

</head>

<body>

<div>內容在此處出現。</div>

</body>

</html>

然後，在將網頁內容貼到 body 裡面取代原本的內容，最後的程式碼如下：

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<meta name="format-detection" content="telephone=no">

<meta name="msapplication-tap-highlight" content="no">

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="stylesheet" type="text/css" href="/Content/jquery.mobile-1.4.5.min.css" />

<link rel="stylesheet" type="text/css" href="/Content/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/css/font-awesome.min.css" />

<script src="/Scripts/jquery-3.2.1.min.js"></script>

<script src="/Scripts/jquery.mobile-1.4.5.min.js"></script>

<script src="/Scripts/bootstrap.min.js"></script>

<script>

$(document).ready(function () {

$("button").addClass("animated bounce");

});

</script>

<title>FreeCodeCamp 測試網頁</title>

</head>

<body>

<div class="container-fluid">

<h3 class="text-primary text-center">jQuery Playground</h3>

<div class="row">

<div class="col-xs-6">

<h4>#left-well</h4>

<div class="well" id="left-well">

<button class="btn btn-default target" id="target1">#target1</button>

<button class="btn btn-default target" id="target2">#target2</button>

<button class="btn btn-default target" id="target3">#target3</button>

</div>

</div>

<div class="col-xs-6">

<h4>#right-well</h4>

<div class="well" id="right-well">

<button class="btn btn-default target" id="target4">#target4</button>

<button class="btn btn-default target" id="target5">#target5</button>

<button class="btn btn-default target" id="target6">#target6</button>

</div>

</div>

</div>

</div>

</body>

</html>

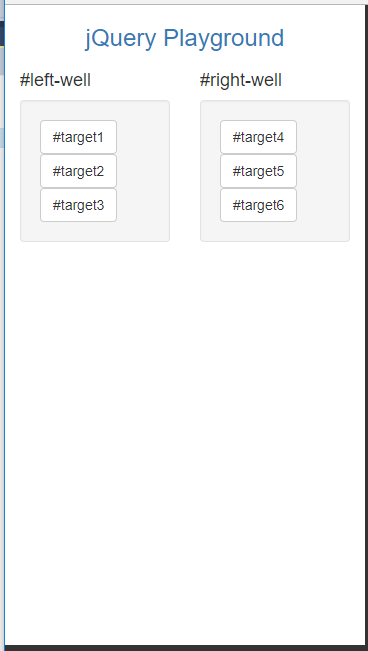
直接執行吧！這時候你會發現，程式執行還是從 index.html 開始，因為VS把整個網站當成一個程式，進入點是 index.html，如果是這樣，我們就在 index.html 裡面加一個鏈結到 freecodecamp.html。

<h3 class="myHeader text-center"><a href="freecodecamp.html">FreeCodeCamp</a></h3>

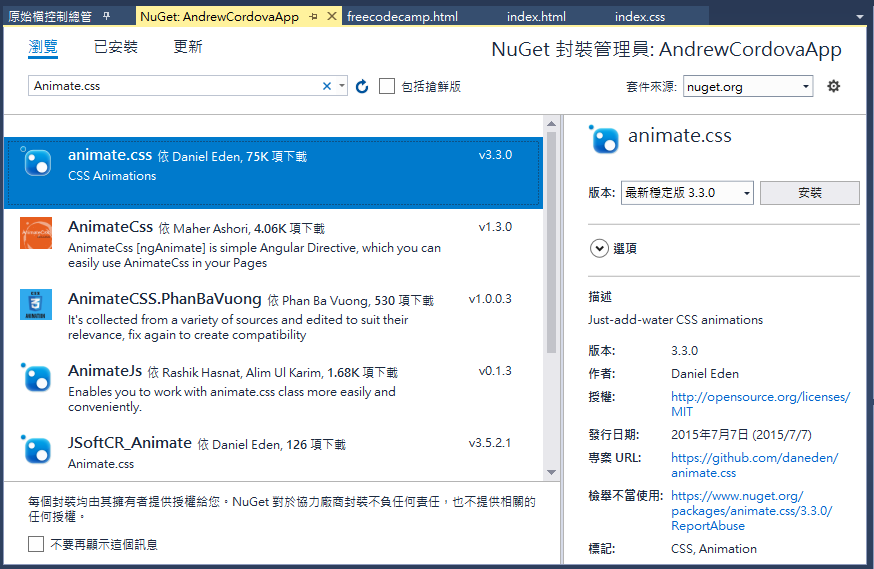
再次執行，先會進入首頁：



點 FreeCodeCamp：



果然跟 FreeCodeCamp 出現的畫面一樣，但是出現一個小問題，就是裡面的按鈕並沒有跳動！原因很簡單，就是因為少了Animate.css 的緣故。快速的進入NuGet 把它安裝好吧：



然後加入這行：

<link rel="stylesheet" type="text/css" href="/Content/animate.min.css" />

恭喜你，按鈕跳動了！由於無法擷取動畫的畫面，只有靠你自己實際執行了。

參考資料，FreeCodeCamp 完整程式碼：

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<meta name="format-detection" content="telephone=no">

<meta name="msapplication-tap-highlight" content="no">

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="stylesheet" type="text/css" href="/Content/jquery.mobile-1.4.5.min.css" />

<link rel="stylesheet" type="text/css" href="/Content/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/css/font-awesome.min.css" />

<link rel="stylesheet" type="text/css" href="/Content/animate.min.css" />

<script src="/Scripts/jquery-3.2.1.min.js"></script>

<script src="/Scripts/jquery.mobile-1.4.5.min.js"></script>

<script src="/Scripts/bootstrap.min.js"></script>

<script>

$(document).ready(function () {

$("button").addClass("animated bounce");

});

</script>

<title>FreeCodeCamp 測試網頁</title>

</head>

<body>

<div class="container-fluid">

<h3 class="text-primary text-center">jQuery Playground</h3>

<div class="row">

<div class="col-xs-6">

<h4>#left-well</h4>

<div class="well" id="left-well">

<button class="btn btn-default target" id="target1">#target1</button>

<button class="btn btn-default target" id="target2">#target2</button>

<button class="btn btn-default target" id="target3">#target3</button>

</div>

</div>

<div class="col-xs-6">

<h4>#right-well</h4>

<div class="well" id="right-well">

<button class="btn btn-default target" id="target4">#target4</button>

<button class="btn btn-default target" id="target5">#target5</button>

<button class="btn btn-default target" id="target6">#target6</button>

</div>

</div>

</div>

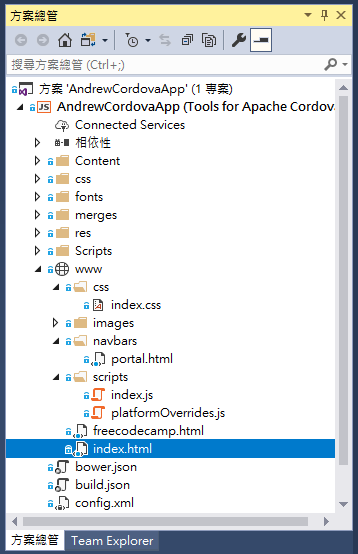
</div>

</body>

</html>

#### 二之六、編輯自己的 index.js

當你利用 Cordova 模塊新增一個方案之後，你會得到一個基本的程式架構，程式的進入網頁是在 www 目錄底下的 index.html。與這檔案相關的還有兩個檔案，分別是 index.css 以及 index.js。參考下圖並熟記其位置：



如果要編寫自己的程式，建議把這兩個檔案都清空，前面的章節有講過如果沒有清空會產生的問題，同時，也舉了 index.css 的例子，現在我們就來談一下如何編輯自己的 index.js。

我們一步一步來，先看一下 index.html 裡面的範例：

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="stylesheet" type="text/css" href="/Content/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/css/font-awesome.min.css" />

<script src="/Scripts/jquery-3.2.1.min.js"></script>

<script src="/Scripts/bootstrap.min.js"></script>

<script src="scripts/index.js"></script>

<script>

$(document).ready(function () {

$('h2').addClass('red-text');

$("#myTest").click(function () {

// Only change code below this line.

$("#myHello").html("你按了input按鈕");

});

$("#login").click(function (event) {

alert("Hello! 你按了 <button> 按鈕\n 要注意：<button> 按鈕有Defaul Event，必須截斷 ");

event.preventDefault();

});

});

</script>

<!-- Only change code above this line. -->

我們一開始在寫 jQuery 的時候，大部分都是用 inline 的寫法，也就是將 script 直接寫在 html 檔案裏面：

<script>

$(document).ready(function () {

$("#myTest").click(function () {

// Only change code below this line.

$("#myHello").html("你按了input按鈕");

});

});

</script>

看一下執行結果：



不過，這樣的寫法似乎有點不太專業，一來你的 html 會太過於龐大，二來 index.js 會空在那裏沒有用處，三來這樣的設計方式就違背了 MVC 的原則，頁面程式與程式碼沒有分開。

要將其分開很容易，只要把原本在 inline 裡面夾在 <script> </script> 中間的程式碼全部移到 index.js 裡面即可，只要寫一行連結：

<script src="scripts/index.js"></script>

Index.js 程式碼如下：

$(document).ready(function () {

$('h2').addClass('red-text');

$("#myTest").click(function () {

// Only change code below this line.

$("#myHello").html("你按了input按鈕");

});

$("#login").click(function (event) {

alert("Hello! 你按了 <button> 按鈕\n 要注意：<button> 按鈕有Defaul Event，必須截斷 ");

event.preventDefault();

});

});

但是，比較成熟的程式設計師會寫成這樣：

$(function () {

$('h2').addClass('red-text');

$("#myTest").click(function () {

// Only change code below this line.

$("#myHello").html("你按了input按鈕");

});

$("#login").click(function (event) {

alert("Hello! 你按了 <button> 按鈕\n 要注意：<button> 按鈕有Defaul Event，必須截斷 ");

event.preventDefault();

});

});

其中 $(function () 就等於 $(document).ready(function () ，只是讓寫法更為簡潔而已，執行的結果一模一樣。

接下來，我們進一步討論 button 與 input 的差異。在這個範例裡面，我們使用了原本 html 所提供的原始 input element，也同時使用的 button element。由於是測試程式，我並沒有處理任何 POST 或是 GET 事件。

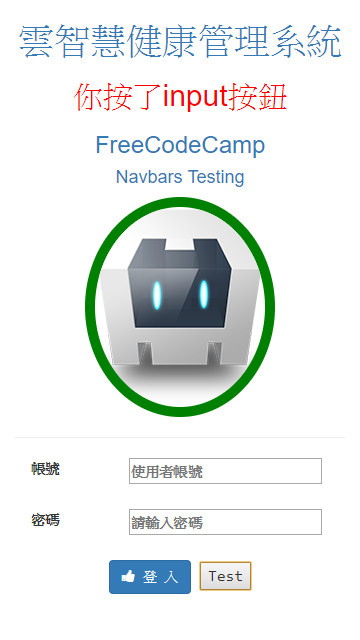
$("#myTest").click(function () {

// Only change code below this line.

$("#myHello").html("你按了input按鈕");

});

當我們按下 input 按鈕之後，如同程式所寫的，會修改 id=myHello 的 html 的內容，執行結果中紅色的字串就是我們顯示出來的內容，如下圖：



但是如果 button 我們也使用同樣的程式碼：

$("#login").click(function (event) {

$("#myHello").html("你按了button按鈕");

});

則執行的結果會變成：



仔細看一下網址的部分，出現 localhost:4400/script/index.js?

這是因為 button element 會有一個 default submit 的動作，會把 form 裡面的資料透過 GET 方式 submit 回 server 端，當我們沒有做任何處理的時候，執行結果就是錯誤的。

要暫時避免這樣的結果，只要在程式碼裡面加一行：

event.preventDefault();

這樣就可以避免錯誤的結果，完整程式碼如下：

$(function () {

$('h2').addClass('red-text');

$("#myTest").click(function () {

// Only change code below this line.

$("#myHello").html("你按了input按鈕");

});

$("#login").click(function (event) {

$("#myHello").html("你按了button按鈕");

event.preventDefault();

});

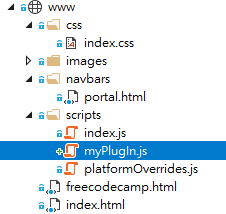
});

#### 二之七、編寫自己的 PlugIn.js

看來 jQuery 你已經用得很熟了，想學習如何自己編寫外掛程式。在這之前，我們先回顧一下上一篇 index.js 的分享。之所以會使用到 index.js，是因為不希望把所有的 script 都寫在 html 裡面，把 script 拆分到 index.js 之後，就等於把介面與程式分離了。

那什麼又是 PlugIn呢? 想想看你在寫 jQuery 的時候，會寫到 $(“h2”).addClass(“red”); 其中的 addClass 就是來自於 jQuery.XXX.js 這個 PlugIn，那是別人寫好的我們拿來用。現在，我們是要自己寫一個 PlugIn 給自己用(當然，別人也可以使用)。如果你想清楚要自己設計外掛程式了，恭喜你，這篇文檔正適合你。用外掛程式和方法來擴展 jQuery 非常強大，把最聰明的功能封裝到外掛程式中可以讓個人及團隊節省大量開發時間。

要编写一个 jQuery 插件，首先請自行新建一個 js 檔案，我就把它命名為 myPlugIn.js，如圖：



在 myPlugIn.js 裡面，需要为 jQuery.fn 对象增加一个新的函数属性，属性名就是插件的名字：

jQuery.fn.myPluginName = function() {

// 外掛程式的具體內容放在這裡

};

或許你看過其他的書！(很多參考書籍會這樣寫：$.fn.) 我們熟悉並鍾愛的 $ 哪兒去了？ 它還在，只是為了確保你的外掛程式不與其它使用 $ 的庫發生衝突，有一個最佳實踐： 把 jQuery 傳遞給 IIFE（立即調用函數），並通過它映射成 $ ，這樣就避免了在執行的作用域裡被其它庫所覆蓋。

(function( $ ) {

$.fn.myPluginName = function() {

// 外掛程式的具體內容放在這裡

};

}( jQuery ));

PS: 跟 index.js 裡面 $(function() { 的寫法比較一下，完全不一樣喔。

現在，在此封閉區域內我們可以隨意使用 $ 替換 jQuery。當我們有了框架之後，就可以開始編寫真正的外掛程式碼了。但在這之前，關於程式內文我要先說明一下。在外掛程式函數的立即作用域中，關鍵字 this 指向調用外掛程式的 jQuery 物件。這是個經常出錯的地方，因為有些情況下 jQuery 接受一個回呼函數，此時 this 指向原生的 DOM 元素。這常常導致開發者在 jQuery 函數中對 this 關鍵字多作一次無必要的包裝。(有時候你會看到 $(this) 的寫法，但是在這裡是沒有必要的)

(function( $ ){

$.fn.myPluginName = function() {

// 沒有必要再作 $(this) ，因為"this"已經是 jQuery 物件了

// $(this) 與 $($('#element')) 是相同的

this.fadeIn('normal', function(){

// 在這裡 this 關鍵字指向 DOM 元素

});

};

}( jQuery ));

$('#element').myPluginName(); //寫在引用處

現在理解了 jQuery 外掛程式的內文以後， 我們來寫一個真正能做點兒事兒的外掛程式。

(function ( $ ) {

//設定成綠色

$.fn.greenify = function () {

this.css("color", "green");

};

}( jQuery ));

這段程式就寫在 myPlugIn.js 裡面，然後，回到我們的 index.html 中連結這個 script，並且使用它：

<script src="scripts/myPlugIn.js"></script>

<script>

$(document).ready(function () {

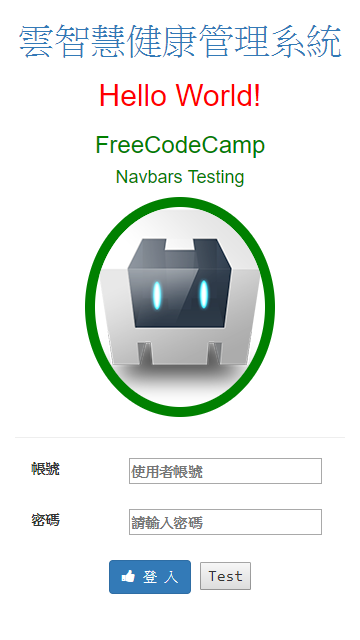
$("a").greenify();

//把所有<a> 連結變成綠色(本來是藍色的)

});

</script>

執行結果如下，FreeCodeCamp 以及 Navbars Testing 兩個連結字串都變成綠色的了：



##### Chaining

我們常會看到 jQuery plugin 在使用的時候，會連接在一起使用，要做到這點很簡單，只要修改程式碼如下就可以：

$.fn.greenify = **function**() {

**this**.css( "color", "green" );

**return** **this**;

}

$( "a" ).greenify().addClass( "greenified" );

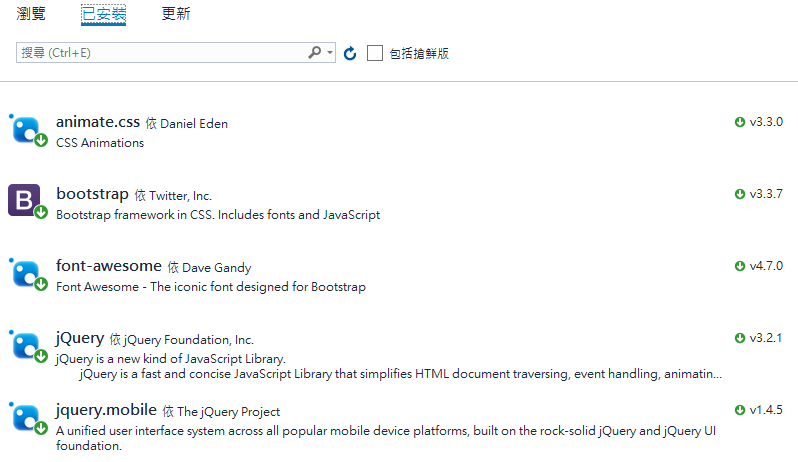
插件的編寫有眾多技巧，一般我都會去下面的網站學習，有興趣可以看看：

<https://learn.jquery.com/plugins/basic-plugin-creation/>

#### 二之八、使用 BootStrap 製作手機APP主選單

首先，先要確定我們的開發方向，就是使用有自適應功能的 BootStrap，一旦使用了 BootStrap，原先的 jQuery.mobile 就不能再使用，因為這兩個 CSS 與 JS 會互相衝突。

之前我們有說明，如何在 Visual Studio 之中安裝NuGet套件，回到我們的方案中，可以看到所有我們已經安裝的套件，如下圖：



安裝套件無所謂，但是在程式中引用的時候就要特別注意，請務必確認在 index.html 裡面，絕對不可以引用 jQuery.mobile.css 以及 .js。請參考程式碼，這是我所引用的 Plugins，jQuery.mobile 相關的都拿掉了，出現的都是相容性沒問題的：

<link rel="stylesheet" type="text/css" href="css/index.css" />

<link rel="stylesheet" type="text/css" href="/Content/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/css/font-awesome.min.css" />

<script src="/Scripts/jquery-3.2.1.min.js"></script>

<script src="/Scripts/bootstrap.min.js"></script>

<script src="scripts/index.js"></script>

<script src="scripts/myPlugIn.js"></script>

主程式是 index.html，執行網頁是會從這裡執行，所以我在主畫面做了一個連結，連到我做的 portal.html，這是我們今天要探討的內容。完整程式如下：

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">

<link rel="stylesheet" href="jquery.mobile-1.4.5.min.css" />

<link rel="stylesheet" type="text/css" href="/Content/bootstrap.min.css" />

<link rel="stylesheet" type="text/css" href="/css/font-awesome.min.css" />

<script src="/Scripts/jquery-3.2.1.min.js"></script>

<script src="/Scripts/jquery.mobile-1.4.5.min.js"></script>

<script src="/Scripts/bootstrap.min.js"></script>

<title>Navbars Testing</title>

</head>

<body>

<nav class="nav navbar-inverse">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="#"><span class="glyphicon glyphicon-cloud-download"></span> 雲智慧健康管理系統</a>

</div>

<div class="collapse navbar-collapse" id="myNavbar">

<ul class="nav navbar-nav">

<li><a data-toggle="tab" href="#home"><span class="glyphicon glyphicon-home"></span> 首頁</a></li>

<li><a data-toggle="tab" href="#menu1"><span class="glyphicon glyphicon-heart"></span> 健檢套餐</a></li>

<li><a data-toggle="tab" href="#menu2"><span class="glyphicon glyphicon-check"></span> 問卷調查</a></li>

<li><a data-toggle="tab" href="#menu3"><span class="glyphicon glyphicon-shopping-cart"></span> 線上下單</a></li>

<li><a data-toggle="tab" href="#menu4"><span class="glyphicon glyphicon-open-file"></span> 健康追蹤</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-log-in"></span> 登入</a></li>

<li><a href="#"><span class="glyphicon glyphicon-log-out"></span> 登出</a></li>

<li><a href="#"><span class="glyphicon glyphicon-user"></span> 個人信息管理</a></li>

</ul>

</div>

</div>

</nav>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<h3>健康管理首頁</h3>

<p>全方位深度健康檢查，個人化全套抗老醫學分析。對健康較重視的高階人士，精心規畫全身性的完整檢查。</p>

</div>

<div id="menu1" class="tab-pane fade">

<h3>健檢套餐</h3>

<p>依每位受檢者選擇之檢查套組而有所不同，請依實際檢查套組為準。</p>

</div>

<div id="menu2" class="tab-pane fade">

<h3>問卷調查</h3>

<p>健康体检自测问卷，依照每個人的問卷結果，提供最適合的健檢項目。</p>

</div>

<div id="menu3" class="tab-pane fade">

<h3>線上下單</h3>

<p>當您確認您所要體檢的項目，可以直接線上預約，並且線上下單付款。</p>

</div>

<div id="menu4" class="tab-pane fade">

<h3>健康追蹤</h3>

<p>您可以在此追蹤您每次在本院檢查的健檢電子報告。本系統會主動建議您的最佳飲食與運動習慣。</p>

</div>

</div>

</body>

</html>

##### HTML基本標籤

我們先從最基礎的開始，在HTML中，我們會使用<ul>來製作表單，ul是unorder list（非順序性清單）的意思，<ul>裡面會搭配<li>來建立清單，像是：

<body>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">Page 1</a></li>

<li><a href="#">Page 2</a></li>

<li><a href="#">Page 3</a></li>

</ul>

</body>

因此，當我們輸入以上內容時，就會有如下的表單：



一般來說，如果我們想要把表單變成水平排列，我們可以在ul的地方改成 <ul class="list-inline">



基本上這是很陽春的介面，我們就可以開始加一些bootstrap的關鍵字來讓它更好看了，我們可以在ul的地方改成<ul class="nav nav-tabs">，像是這樣：

<ul class="nav nav-tabs">

<li><a href="#">Home</a></li>

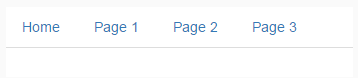
<li><a href="#">Page 1</a></li>

<li><a href="#">Page 2</a></li>

<li><a href="#">Page 3</a></li>

</ul>

這時候，結果就會變成如下：



這就是基本的tabs了，如果我們要讓某個tabs看起來像是目前的頁面，可以用<li class="active">：

<body>

<ul class="nav nav-tabs">

<li class="active"><a href="#">Home</a></li>

<li><a href="#">Page 1</a></li>

<li><a href="#">Page 2</a></li>

<li><a href="#">Page 3</a></li>

</ul>

</body>



##### 可下拉式的標籤（dropdown tabs）

為了製作可下拉式選單，我們會用到的關鍵字包含許多，整個程式碼如下：

<ul class="nav nav-tabs">

<li><a href="#">Home</a></li>

<li><a href="#">Page 1</a></li>

<li><a href="#">Page 2</a></li>

<li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Dropdown<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#">First</a></li>

<li><a href="#">Second</a></li>

<li><a href="#">Third</a></li>

</ul>

</li>

</ul>

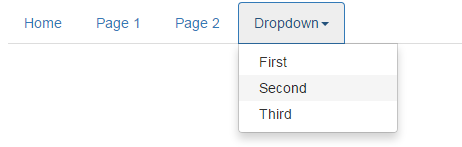
<li class="dropdown">：其中，我們要在希望增加下拉式選li的地方加上這段，以建立一個超連結，而這個超連結的class是<class="dropdown-toggle">。

<data-toggle="dropdown">：這個data-toggle是用來使用javascript的，讓網友點擊該超連結時，會出現下拉式選單，如果沒有這段，不管怎麼點，下拉式選單都不會出現。

<span class="caret">：這是用來產生「下三角形」的符號，用來暗示網友這是下拉式選單，沒有這個的話，下三角形還是在，只是網友點擊之後，才知道這是下拉式選單。

<ul class="dropdown-menu">：接著，要建立一個新的 <ul>標籤，這個標籤的裡面，放的就是下拉式選單的內容，裡面記得要加上「dropdown-menu」，沒有這段的話，下拉式選單不會隱藏在主選單當中。

結果就會像這樣：



##### 動態性標籤（Dynamic Tabs）

這個的作法，是讓網頁中文字的內容，可以隨著點選的tab而動態（淡入淡出）的效果，這樣的效果同樣會用到很多不同的語法：

<ul class="nav nav-tabs">

<li><a data-toggle="tab" href="#home">Home</a></li>

<li><a data-toggle="tab" href="#menu1">Page 1</a></li>

<li><a data-toggle="tab" href="#menu2">Page 2</a></li>

<li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">Dropdown<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="#">First</a></li>

<li><a href="#">Second</a></li>

<li><a href="#">Third</a></li>

</ul>

</li>

</ul>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<h3>HOME</h3>

<p>Some content.</p>

</div>

<div id="menu1" class="tab-pane fade">

<h3>Menu 1</h3>

<p>Some content in menu 1.</p>

</div>

<div id="menu2" class="tab-pane fade">

<h3>Menu 2</h3>

<p>Some content in menu 2.</p>

</div>

</div>

data-toggle="tab"：這會產生相對應的javascript語法，如果沒有這個的話，點標籤會沒有任何反應，所以每一個我們希望點了會有反應的標籤，都要加上這段語法。

在超連結的地方，我們可以看到，超連結網址的名稱，要和下面內容id的名稱一致，第一個標籤會連到id="home"。

class="tab-content"：這個的目的是要讓下面不同div呈現的內容都會在同樣的位置，如果沒有這個的話，因為三個div是有順序的，所以位置會變成由上往下來排（不會在相同位置處呈現），加上這個之後，等於讓每個div的位置，都是出現在最左上角。

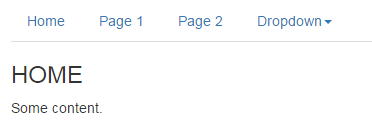
tab-pane：tab-pane和上面tab-content的效果很相似，沒有這個的話，div內容的位置也會出現問題。

fade：以淡入、淡出的方式呈現，沒有這個的話，就會像超連結一樣，點了直接消失、出現。

in：意思是先淡入（顯示）的意思，通常是用在首頁，如果沒有這個in的話，網友一開始進到home這個頁面時會沒有任何內容，因為home這個頁面還沒有淡入（顯示）。

active：搭配in一起使用，表示這是一開始就要活動的頁面，如果沒有這個的話，網友一開始進到home這個頁面時也不會有任何內容。

結果就像這樣，動畫的效果可以到W3CSchool觀看：



##### 適應性網頁的導覽列

在適應性網頁設計中（RWD），當網頁寬度太小時，這些導覽列會直接縮成一個按鈕（通常長的像「三」），我們一樣可以寫出這樣的語法來：

<nav class="nav navbar-default">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="#">Brand</a>

</div>

<div class="collapse navbar-collapse" id="myNavbar">//底下放要縮放的內容

<ul class="nav navbar-nav">

<li><a data-toggle="tab" href="#home">Home</a></li>

<li><a data-toggle="tab" href="#menu1">Page 1</a></li>

<li><a data-toggle="tab" href="#menu2">Page 2</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Log in</a></li>

<li><a href="#">Log out</a></li>

<li><a href="#"><span class="glyphicon glyphicon-user"></span> Admin</a></li>

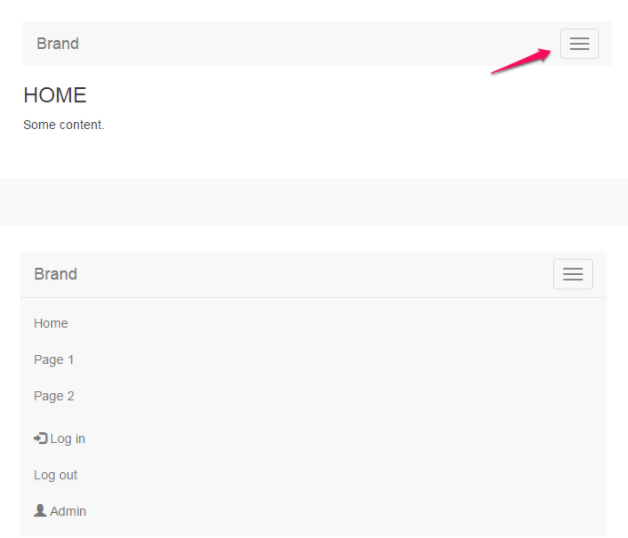
</ul>

</div>

</div>

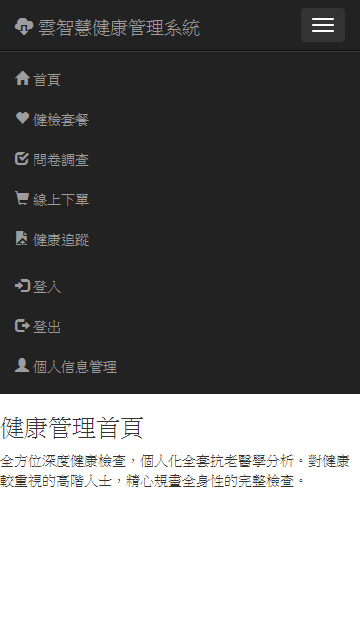
</nav>

這段程式碼當我們把網頁視窗縮小的時候，會顯示出如下的效果：



把我們的程式稍加修改一下，就會呈現下面的效果：







### 三、MS SQL 资料库操作指南

這裡記錄了MS SQL Server 的操作方式，以及需要各位Step by step 修改資料庫的步驟

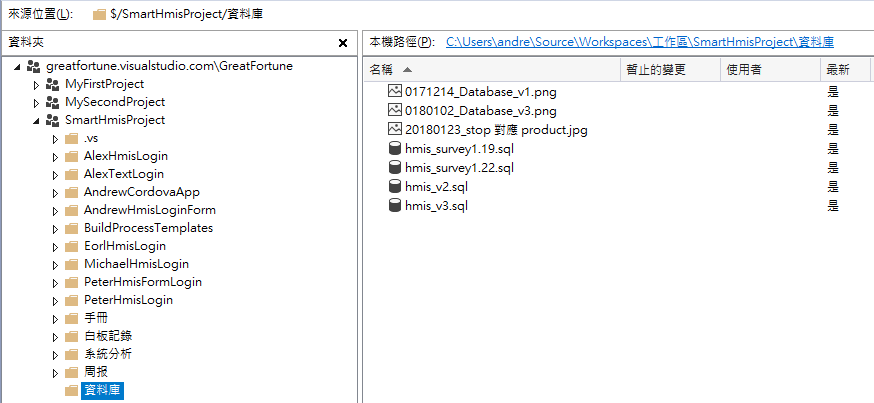
#### 三之三、原始資料庫(請確認已達成)

在開發雲智慧之前，由於資料庫也隨著開發的腳步而修正，所以我們必須了解資料庫版本的變化。在初期，我們先行導入了 hmis SQL Server 資料庫，相信每個人都已經完成此步驟，但是後續會有很多變動，也會在某個時間點將最新版的資料庫重新導入，所以在此記錄所有資料庫的變動。

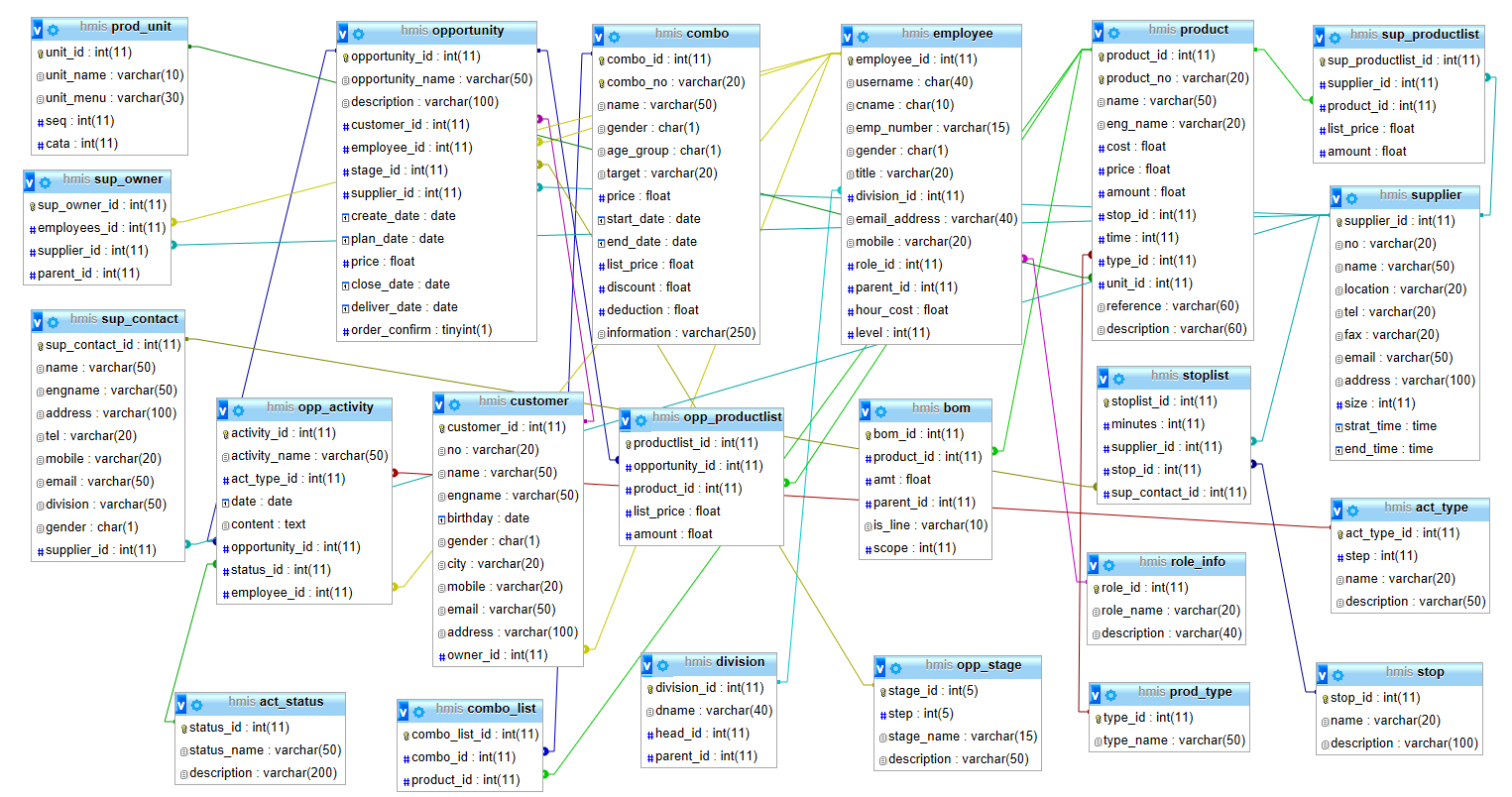
首先，在此說明我們的資料庫的演化過程。

##### 原始MySQL hmis 資料庫

原始的 hmis 資料庫是由mes, crm+scm 兩大資料庫整併修改而來，所以原始資料庫是 MySQL 版本，裡面已經有程式開發所需要的範例資料，該資料庫存放於 Team Service 的資料庫目錄中：

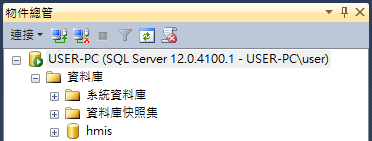


檔案名稱為 hmis\_v3.sql，如果各位覺得使用 MySQL 比較順手，可以導入此資料庫到 MySQL 之中參考。此原始資料庫的資料表關聯圖如下：



##### 原始 SQL Server hmis 資料庫

由於我們使用Visual Studio 2017 做為開發工具，而其搭配的主要資料庫為 Microsoft SQL Server，再加上 SQL Server 未來的擴充性比較適合商業用途，因此我們把健康管理系統的標準資料庫訂為 MS SQL Server 2014(當然，你也可以安裝更新的版本，但是電腦中有舊版本的時候，安裝新版本要務必小心相容性)，所以，我將原本的MySQL hmis 資料庫做了一次 Migration (抱歉，不知道中文怎麼說)，使用 Microsoft SQL Server Migration Assistant for MySQL 這個工具，在 Wiki 3-2 的分享中，我已經介紹了如何使用此 Tool 來做轉換。轉換後的 SQL Server hmis，相信已經都放到各位自己 SQL Server 裡面了。如下圖：



SQL Server 的資料庫檔案有兩個，分別是 .mdf 與 .ldf，它們沒有辦法放到 Team Service 的目錄裡面分享給大家。所以如果有新版本的異動的時候，我會放在 ftp Server 提供大家下載，並說明正確的卸載與附加的方式。

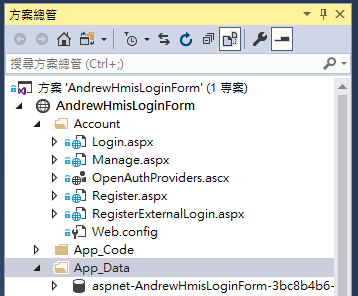
##### AppData Login 原始資料庫

目前我們的系統是使用 WebForm Template 為原型而開發的系統，而這個 Template 裡面，原本就存在使用者註冊、登入、管理帳號密碼的功能。在我們使用內建的註冊功能的時候，會自動產生一個 Login 的原始資料庫，但是這個資料庫絕對不可以複製，因為每個人的開發環境的位置不同，系統所建立的資料庫也會不同。

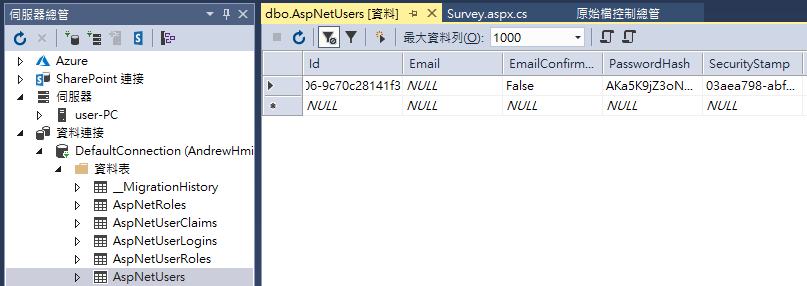
如下圖，我相信大家都有新增自己的 HmisLogin Web 程式：



請打開自己的(不要開別人的) 專案，然後執行此專案，如果是第一次使用，請進入註冊畫面註冊一個測試帳號。這時，系統就會產生一個 Login 資料庫，請到方案總管裡面打開 AppData 這個目錄，如下圖：

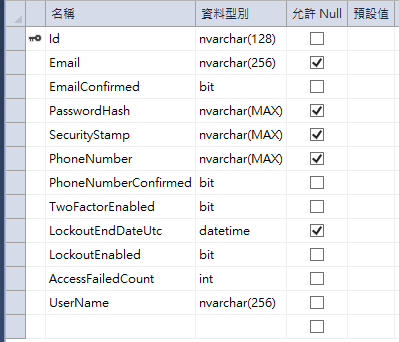


這個資料庫的名稱就會叫做 aspnet-\*，後面有一串 GUID 的編號命名，每個人都會不一樣，請不要對此資料庫做任何的變動，但是我們可以看一下它的內容：



在伺服器總管中打開 DefaultConnection ，並打開 AspNetUsers ，裡面就會看到一筆剛才註冊的使用者記錄，請特別注意這筆記錄的 id 欄位，這是一個 GUID 的欄位，就跟我們平常設計資料表都會放一個自動遞增的主鍵一樣，它是用 GUID，各有其利弊，我們自己的資料庫還是繼續使用自動遞增。

這裡只有三個欄位我們知道一下就好，分別是 Id, UserName, PasswordHash，但是我們千萬不要自行變動他們的內容，否則會出問題。如下圖：



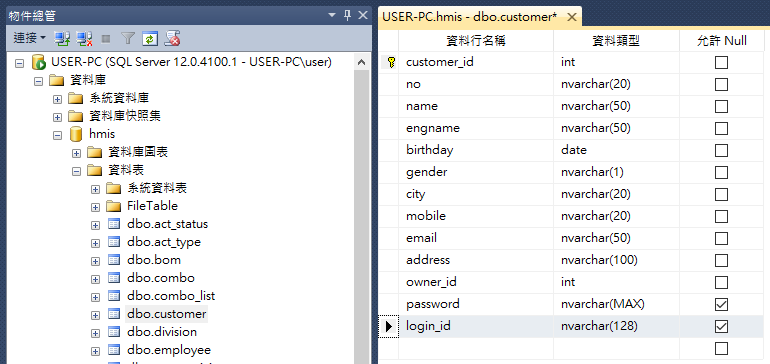
UserName 是使用者在註冊的時候自己取的名字，通常是不能有特殊符號的英文字，我知道在大陸很喜歡使用手機號碼當成ID，但是在台灣或是其他國家還是會以英文名字為主。PasswordHash 是使用者自己輸入並且再次確認的密碼，這裡有加密過。Id 則是系統產生的唯一編號。

之前 Peter 有試著將這個資料庫整併到我們的 hmis 資料庫裏面，但是還沒有成功，所以我們暫時決定，讓這個資料庫獨立存在。但是我們要如何利用此資料庫呢?

針對外網系統而言，hmis 資料庫裏面有一個 customer 資料表，是讓使用者註冊用的，現在我們用 VS 提供的內鍵註冊功能，只要在使用者註冊之後，把這裡自動生成的ID讀出來，然後在 customer 裡面多加一個 login\_id 的欄位，新增一筆記錄，將此ID填入 login\_id 欄位即可。

##### SQL Server hmis PETER 修訂版

就是因為如此，我們的 SQL Server hmis 會有一個小小的修訂，就是在 customer 裡面，多了兩個欄位，一個是 Peter 用來存放 GUID 的 login\_id，一個是 Alex 用來測試的 password。如下圖：



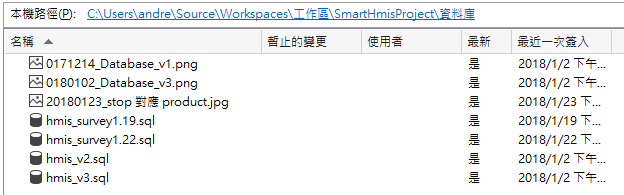
欄位的長度，我們都是比照 AppData 中的 AspNetUser 來放，分別是 nvarchar(128) 與 nvarchar(MAX)。

要特別注意，如果你的 customer 資料表裡面沒有這兩個欄位，可以自行加入，因為是小改版，我們沒有特別匯出讓大家匯入。這是因為我們還沒有做最後的定案。特別說明一下，內網的部份我們也會使用類似外網的 Login 功能，但是不提供註冊功能，而且內網使用的 hmis 資料表是 employee，目前沒有變動。

##### 原始 MySQL hmis\_survey 資料庫

外網有一個重要的功能，就是可以讓使用者在線上做問卷調查，做出來的結果經過分析之後，我們會自動給予適當的健檢套餐。問卷調查是由 Michael 在大陸衛生部門的官方網站下載的標準問卷，我們一來為了方便問卷題目在未來做變更，二來為了讓問卷調查網頁呈現動態展示，因此由 Michael 先行設計了一個問卷調查的資料庫 hmis\_survey，由於我們之前的資料庫都是使用 MySQL，因此這個 hmis\_survey 的原型是 MySQL 的版本。

此版本目前經過我和 Michael double check，已經是最後版本，之前如果有人下載到舊的版本，請重新匯入。進入 Team Service 的資料庫目錄，hmis\_survey1.22.sql 為最新的本版，如下圖：



拿到 MySQL 版本之後，請先匯入 MySQL 資料庫，如下圖：



這個版本最重要的部分，就是 person 資料表，此資料表包含了300個欄位，這是完全針對標準問卷調查表而設計與切割出來可以符合大數據分析的資料表，未來可以透過 Python 編寫線性回歸，SVM分析監督學習，或是聚類特性的無監督學習。

欄位名稱的命名原則如下，透過下面 SQL 語言可以產生問卷分析總表。

SELECT `category\_id`,`category\_name`,`question\_id`,`question\_name`,`content\_no`,`content\_name`, `ctype\_name`

FROM `answer`

LEFT JOIN `question` USING(`question\_id`)

LEFT JOIN `category` USING(`category\_id`)

LEFT JOIN `content\_type` USING(`ctype\_id`)

WHERE 1

ORDER BY `category\_id`,`question\_id`,`answer\_id`



總表的筆數為543筆記錄，這裡面會有 question\_id 已經 content\_no，將這兩個欄位相加就形成 person 的欄位名稱，然而欄位又有四種型態，分別是文字、數字、Checkbox、與 Radio button，當型態為文字、數字或是單選按鈕的時候(或者是由三個數字組合成的日期，如生日)，答案只有一個，所以答案欄位為 Q1(姓名:文字), Q2(性別:答案為A或B), Q3(出生日期:日期)等。但是答案如果是複選按鈕，如下圖：



第13題的答案欄位就是 Q13A, Q13B, ……, Q13G(肥胖症), Q13H(慢性腎臟疾病), …… , Q13N(精神疾病), Q13Z(其他)。

這裡有兩個特別規定，一是 “其他” 原本是 “O”，但是我們一律把 “其他” 定義成 “Z”，這是因為預防未來如果新增一個疾病O的時候，也不會把其他選項擠到P而改變了欄位的定義而造成數據分析時候的失真。二是如果填問卷者在項目打勾的時候，我們就在該欄位存 “V”，沒填則保持空白。

Person Table 裡面的 customer\_id 則是與 hmis.customer 連接的外部鍵，然而目前暫時不與 hmis 整併。

##### 原始 SQL Server hmis\_survey 資料庫

匯入 MySQL 資料庫之後，再透過 Migration Tool 轉換成 SQL Server hmis\_survey 資料庫。如下圖：



目前這部分由 Peter 開發中，所以其他人如果也會使用到，請自行由 MySQL 轉換。暫時不提供 mdf 與 ldf 檔案。在未來此資料表會整併到 hmis 資料庫。

#### 三之四、資料庫的小改版

由於開發過程中，所有人都使用自己的資料庫，而這個資料庫都是一開始的時候導入的，後面就會發生問題，例如當 Alex 在寫 Stop 站點功能的時候，會新增一些資料到資料表裡面，Peter 會新增問卷調查的資料，因此資料庫就不同步了，但是我們又沒有辦法每天更新資料庫，那會造成更大的麻煩，所以我就將一些小改版的步驟寫下來，大家直接在 Visual Studio 的整合環境裡面修改資料即可。這樣也可以讓大家熟悉 Visual Studio 的操作。

##### Stop 資料表

首先解釋 stop 資料表的作用，我一共定義了十個檢查站點，如下圖：



這代表了如果某家醫院提供了所有的體檢項目，就必須有這十個站點才能完成這些項目，如果醫院沒有B超室，那麼至少有四項檢查是沒有辦法做的，在我的系統分析定義中，有每個站點可以做的檢查，如下圖：



這些都是原始範例，之後也有可能變動，裡面還記錄的每種檢查所需要的時間，這些在資料庫裏面都還沒有實現。

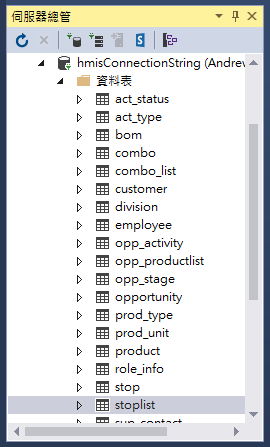
不過目前 stop 資料表先不做變更，我們先將這十個站點定案。

##### Stoplist 資料表

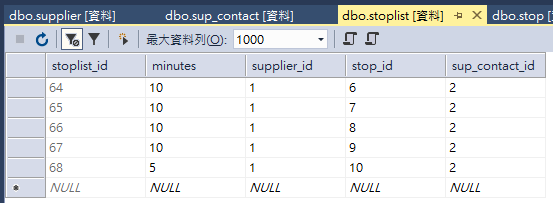
站點確認之後，接下來要做的，就是定義每個醫院所擁有的站點。為何要這樣做？由於我們的系統是一個通用系統，系統可以支援多家不同的分院同時運作，既然是不同的分院，就有可能每家醫院所提供的站點都不一樣。

要達成這樣的功能，就必須透過 stop\_list 來完成，這個資料表紀錄了 supplier\_id (醫院)、stop\_id(站點)、以及 sup\_contact\_id(站點坐鎮的醫生)、最後是 minutes (檢查時間)。

在我的範例資料裡面，每間醫院都只有五個站點，是不完整的，所以請大家坐下面的操作，在伺服器總管中展開 hmis 資料庫，選取 stoplist 資料表，按滑鼠右鍵，選擇[顯示資料表資料]，如下圖：

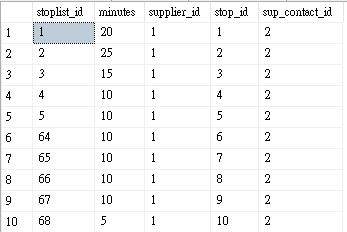


然後在表格裡面把第一家醫院的 6 7 8 9 10 站點資料加進去。如下圖(請完全按照下面每個欄位的資料填入，這樣才會一致)：



PS: 目前 Alex 正在製作 stoplist 的編輯功能，所以 Alex 的版本可能會比我們現在做的修正多，其他人只要新增這五筆資料就好。

##### Minutes 欄位說明



我們先看一下上圖，目前第一家醫院的10個站點所需要的時間，都記錄在 stoplist 裡面。我們以第二個站點為例，在系統分析資料裡面，B超(如下圖)其實細分為腹部彩超15分鐘、女性盆腔彩超10分鐘、颈动脉彩超5分鐘、前列腺彩超5分鐘，但是上面的表只抓了25分鐘(以最長時間為準)，這是我暫時做的簡化，抓一個平均時間，這樣就不必每次都按照檢查的細項來計算。

然而婦科與外科檢查，理論上最長時間為15分鐘，但是這裡在實際運作時，醫生可以壓縮在十分鐘內完成，所以還是簡化為十分鐘。



當然，未來如果有所更動，會一次性修正資料庫架構。

##### 套餐資料修正

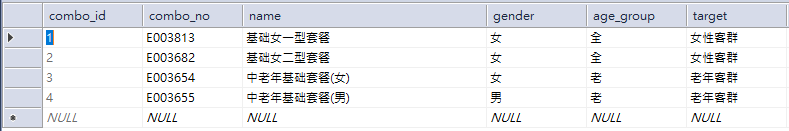
在我們的範例套餐裡面，有四個套餐，分別是年輕女一女二、與老年男女，這個範例會有問題，少了一個年輕男，為了在開發初期不要更動資料庫，請大家按照下面的步驟將年輕女二改成年輕男。

首先請參考下圖，是年輕女二的系統分析表：

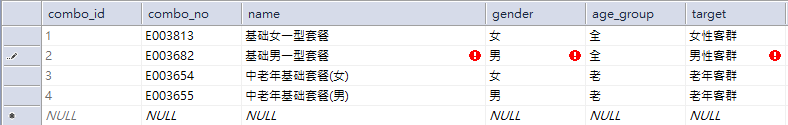


紅色的部分就是我們要修正的地方，但是這個修正橫跨多個資料表，所以請務必小心。

##### Combo 資料表



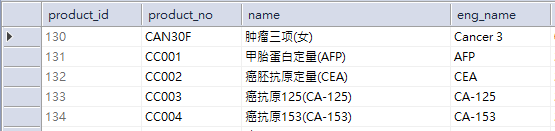
選取 combo 資料表按滑鼠右鍵，選擇[顯示資料表資料]，會出現上圖，然後將第二筆資料的 name, gender, target 改成下圖：



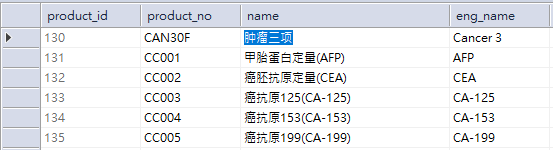
將游標移動到第三行，此資料便修改完成，紅色驚嘆號消失。

##### Product 資料表

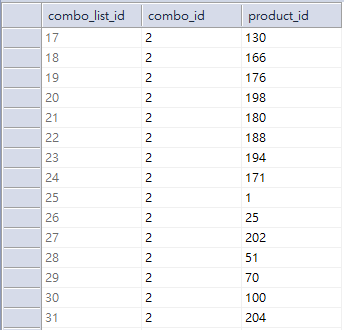
找到 product\_id = 130 那行，會看到 “肿瘤三项(女)”，由於這個項目是男女通用，所以請將其改成 “肿瘤三项” 即可。



修正後：



##### Combo\_list 資料表



這是原本 combo\_id = 2 的所有資料，我們要刪除婦科常規(product\_id = 176)，外科檢查(女)改為外科檢查(男) (product\_id = 180 改為 220)，女性盆腔彩超改成前列腺彩超 (product\_id = 194 改為 223)

修改後如下圖：請確認完全正確！



完成以上的步驟，小改版就全部完成，請務必確實操作。

#### 三之五、資料庫的資料新增

其實在每次資料庫小改版的時候，讓大家自行動手修改是有目的的，我希望透過實際操作，可以更了解資料庫的結構，這樣有助於日後系統開發的速度。所以，我會清楚說明架構讓大家了解。

接下來我要說明體檢項目與檢查站點之間的關係，並透過實際資料的新增完成更完整的範例資料庫。

所謂的站點，就是醫院或是健檢中心裡面的檢查室，下圖是我初步設計的資料，總共有十個站點，就代表醫院有十間檢查室。



這些檢查室是我按照在台灣體檢的經驗設計的，照心電圖的時候會有一間獨立的檢察室，照X光當然有X光室，一進去體檢中心要量身高體重血壓等會有一間測量室，牙科、眼科和婦科因為設備比較特別，所以有獨立的小檢查室，採血會有一個抽血櫃台，順便讓病人將尿液、糞便檢體放在那裡，所以我放在同一個檢查室。至於檢查時間，我也是自己估計的，抽血需要時間，但是驗血是後製作業，所以我時間放0，這就是站點設計的邏輯。

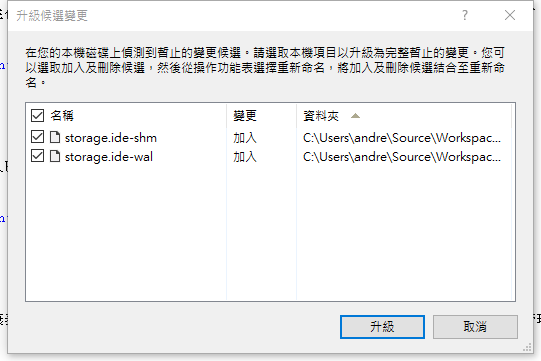
但是如果健檢項目增加，站點就會增加，例如，如果增加了內科與耳鼻喉科的檢查，又假設內科與耳鼻喉科共用同一個檢查室，那麼就需要增加一個內科檢查室的站點，並且將內科與耳鼻喉科這兩個體檢項目與估計的時間加入，如下圖：



##### 新增 stop 資料

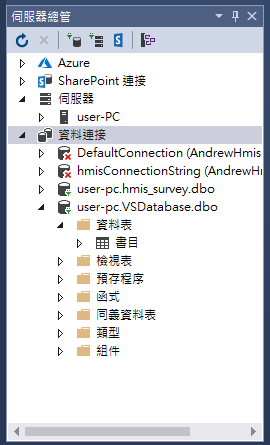
目前這個時間點，Alex 正在設計內網的站點資料維護，在寫程式的時候，一定會測試新增修改刪除功能，所以資料表的內容會有變化。不過沒有關係，其他人還是要按照我的新增過程修改自己的資料庫。Alex 不同的資料內容只要最後匯入此範例資料庫就行了。

開始之前，我先分享一下Visual Studio 的使用經驗，由於我同時要編寫好幾個方案，而每個方案都使用不同的資料庫，再加上我每天都會同步與遷入方案資料(我建議大家都這樣做)，所以第一個最常出現的狀況就是，會產生很多 “排除的變更” ，排除的變更裡面最常出現的就是跟資料庫相關的檔案，如下圖：



這些檔案並不是我們去編輯的，而是系統在執行的時候產生的，所以被放在排除的變更裡面，我的習慣是，直接將這些檔案升級，讓他們變成 “包含的變更”，然後一併遷入 Team Server就會解決總是在排除的變更裡面有幾千筆變更的狀況。這時候，就很有可能出現衝突了，建議如果是這種狀況，就先退出 Visual Studio讓系統停止執行，然後再進去遷入一次，通常就可以正常簽入了。

第二個常見狀況就是有時候我們那些storage.ide檔案沒有處理好，就會造成我們原本在伺服器總管裡面連結好的資料連接消失了。這時候千萬不要重新連結，只要開啟使用該資料連接的方案，資料連接就會回來，如下圖：



第三個狀況是執行時候的問題，方案原本可以正常執行的，但是因為切換了好幾個方案個別編寫程式，可能就造成一些環境變數的改變，當回到之前的方案在執行的時候，就會出現類似下圖的錯誤：



狀態不合法並不代表程式有錯，只要重啟 Visual Studio 再執行一次，通常就會正常了。

言歸正傳，我們要新增一筆站點資料，請先顯示 stop 資料表資料：

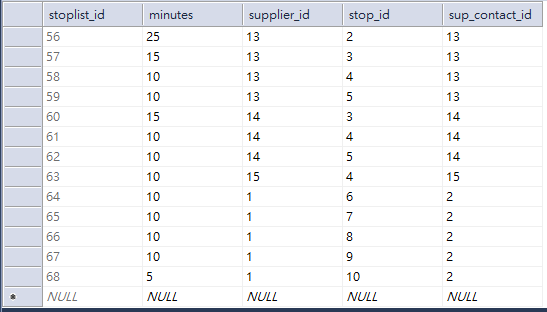


然後請在第11行將名稱與說明複製進去，name = “內科檢查室”，description = “內科與耳鼻喉科共用檢查室”，完成之後如下圖，請注意 stop\_id 目前到11，如果你因為做了刪除動作造成這個 stop\_id 不對，那麼後面就會有問題，如果你的 id 超過 11，那麼至少把第11筆資料改成與這裡相同：



##### 新增 stoplist 資料

同樣的，由於新增了一間檢查室，我們至少要讓第一家醫院擁有所有的檢查室，所以接下來請再 stoplist 裡面新增一筆資料，原始資料表如下：



請加入第69筆資料，假設平均時間是12分鐘，第一家醫院 supplier\_id = 1，醫生 sup\_contact\_id = 2，新的 stop\_id = 11：



移動游標即新增完成。

##### 新增 product 資料(練習)

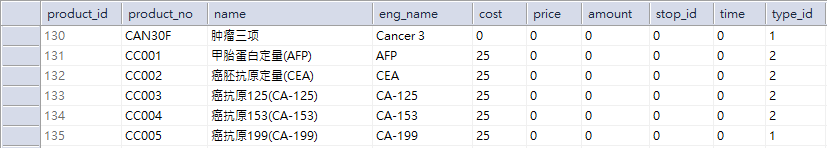
接下來就是重頭戲，我們要新增一個套餐，與套餐相關的資料表有 product: 記錄父項目(level 1) 與子項目(level 2)，之前的章節有解釋過，父項目如同成品，是客戶可選擇的項目，子項目如同原料，是屬於父項目的內容，客戶不能選擇。有一個例外是，如果這個子項目本身就是一個獨立的檢查項目的時候，就要從bom資料表裡面去判定了。

為何我們的標題上面有一個(練習)？因為這個工程太大，新增項目太多，我實際測試的結果，只要新增任何一筆出現失誤，id就會跳號，所以每個人做出來的 product\_id 一定會不一致。到時候資料庫就全亂了。因此，想要練習的可以練習，最後的範本資料庫在過年前我們會重新統一匯入一次。

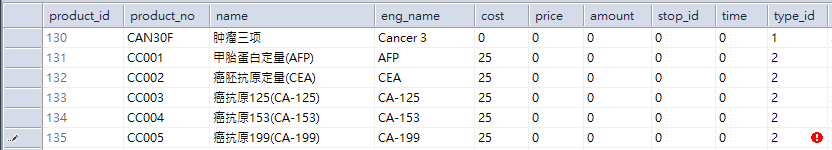
新增之前，我們先用一個系統分析的健檢套餐範例來說明，“基础男一型套餐”，如下圖：



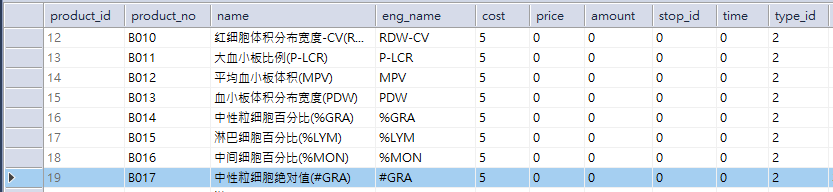
這裡面每一筆資料，都是一個 product，腫瘤三項(CAN30F) 的 product\_id = 130，這屬於父項目，type\_id = 1，而裡面有三個子項目(CC001, CC002, CC003) 的 type\_id = 2。如下圖：



請特別修改一下，這裡面 CC005 我的 type\_id 放錯了，請改成 “2”。修正後如下圖：

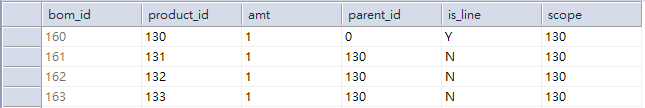


另外還有一個錯誤，Product\_id = 19，type\_id 由3改成2。修正後如下圖：



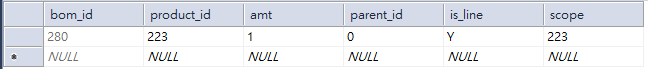
當我們要把父子項目連結起來的時候，則是再透過 bom 資料表來做，而無法只用 product 一個資料表來達成，這是因為不同的父項目，可能會有重複的子項目。關於這點我們在MES系統裡面的BOM表已經討論過。

接續腫瘤三項(CAN30F)的例子，在bom 表裡面，就必須儲存如下：

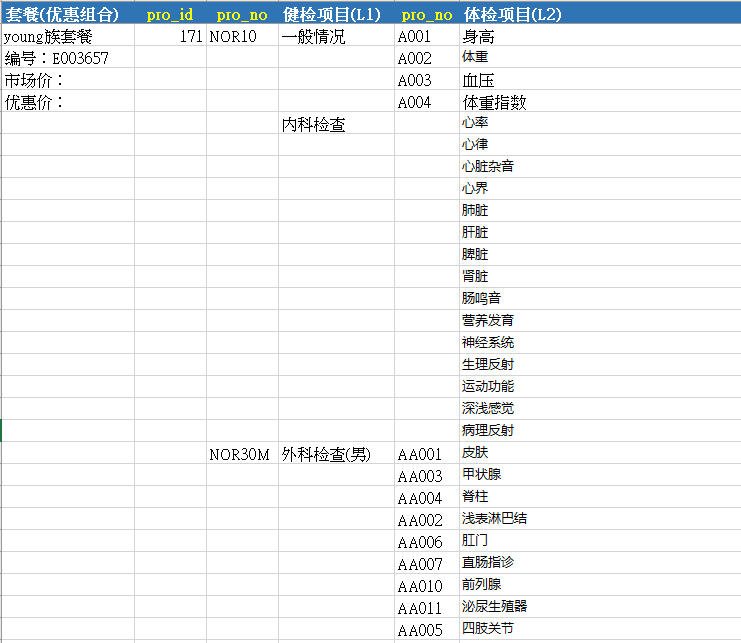


Parent\_id = 0 就是父項目，所有的子項目的 parent\_id 指向這個父項目的 product\_id，is\_line 表示是否可以讓客戶選取，scope 原則上記錄父項目的 product\_id，代表他們是一組的。

如果是單項，如：(AA301)前列腺彩超，它在 product 資料表裡面依舊是個子項目(因為未來可能會被其他父項目包含)，但是在 bom 表裡面會單獨存在，代表是一個單項可選取的項目，如下圖：



明白以上的解釋之後，我們再來實際新增一個套餐 “young族套餐（男）”。下圖是Michael整理的部分表格，有些項目是已經存在的，有些則是新的項目，我們就從新的項目開始：



###### 內科檢查

上圖中第一個新增的項目就是內科檢查，為了方便各位輸入，我把文字列在下面，其中父項目輸入六個欄位，子項目輸入五個欄位，其他文字欄位把NULL刪除，數字欄位輸入0 (unit\_id =35 為布林值)：

NOR30A 内科检查 stop\_id = 11 type\_id = 1 unit\_id=35 description = 筛查內科疾病，对有异常的初步检查。

AB001 心率 stop\_id = 0 type\_id = 2 unit\_id=35

AB002 心律 stop\_id = 0 type\_id = 2 unit\_id=35

AB003 心脏杂音 stop\_id = 0 type\_id = 2 unit\_id=35

AB004 心界 stop\_id = 0 type\_id = 2 unit\_id=35

AB005 肺脏 stop\_id = 0 type\_id = 2 unit\_id=35

AB006 肝脏 stop\_id = 0 type\_id = 2 unit\_id=35

AB007 脾脏 stop\_id = 0 type\_id = 2 unit\_id=35

AB008 肾脏 stop\_id = 0 type\_id = 2 unit\_id=35

AB009 肠鸣音 stop\_id = 0 type\_id = 2 unit\_id=35

AB010 营养发育 stop\_id = 0 type\_id = 2 unit\_id=35

AB011 神经系统 stop\_id = 0 type\_id = 2 unit\_id=35

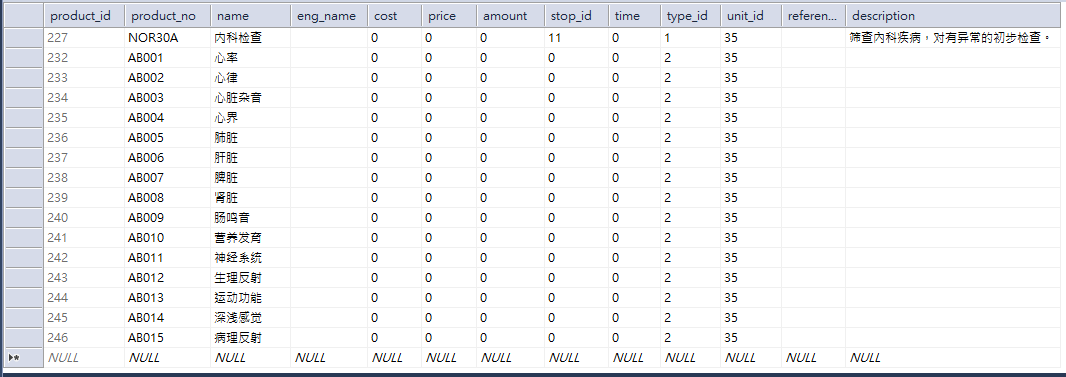
AB012 生理反射 stop\_id = 0 type\_id = 2 unit\_id=35

AB013 运动功能 stop\_id = 0 type\_id = 2 unit\_id=35

AB014 深浅感觉 stop\_id = 0 type\_id = 2 unit\_id=35

AB015 病理反射 stop\_id = 0 type\_id = 2 unit\_id=35

輸入完畢結果如下：



如果你的 product\_id 跟我不一樣，因為是自動產生的，只有一招了，刪除多出來的id，直接修該既有id上的資料。

確定 227對到 NOR30A，232對到 AB001以下，那就可以繼續往下了，不然都會是錯的。

###### 耳鼻喉科檢查



上圖紅色的部份是我新加入的耳鼻喉科的編號，同樣的，我把文字列在下面：

ENT10A 耳鼻喉科检查 stop\_id = 11 type\_id = 1 unit\_id=35 description = 筛查耳鼻咽部疾病，对有喉部症状或发声异常的初步检查。

AC001 听力 stop\_id = 0 type\_id = 2 unit\_id=35

AC002 外耳道 stop\_id = 0 type\_id = 2 unit\_id=35

AC003 耳廓 stop\_id = 0 type\_id = 2 unit\_id=35

AC004 鼓膜 stop\_id = 0 type\_id = 2 unit\_id=35

AC005 鼻窦 stop\_id = 0 type\_id = 2 unit\_id=35

AC006 鼻中隔 stop\_id = 0 type\_id = 2 unit\_id=35

AC007 鼻甲 stop\_id = 0 type\_id = 2 unit\_id=35

AC008 鼻道 stop\_id = 0 type\_id = 2 unit\_id=35

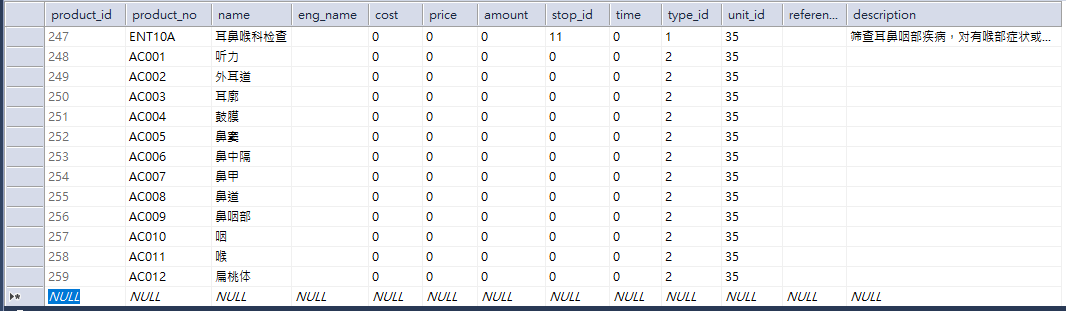
AC009 鼻咽部 stop\_id = 0 type\_id = 2 unit\_id=35

AC010 咽 stop\_id = 0 type\_id = 2 unit\_id=35

AC011 喉 stop\_id = 0 type\_id = 2 unit\_id=35

AC012 扁桃体 stop\_id = 0 type\_id = 2 unit\_id=35

完成如下：



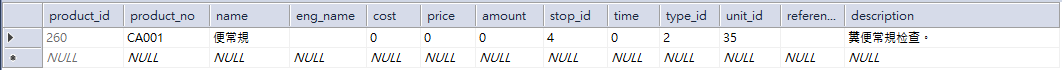
###### 便常規



這是一個單項檢查項目，只需要輸入一個子項目即可。

CA001 便常规 stop\_id = 4 type\_id = 2 unit\_id=35 description = 糞便常規检查。

輸入結果如下：



###### 微量元素5项



其中 unit\_id = 31 代表數值。

EMT50 微量元素5项 stop\_id = 4 type\_id = 1 unit\_id=31 description = 身體铁(Fe)、镁(Mg)、铜(Cu)、钙(Ca)、锌(Zn)微量元素檢測。

EM001 铁(Fe) eng\_name=Fe stop\_id = 0 type\_id = 2 unit\_id=31

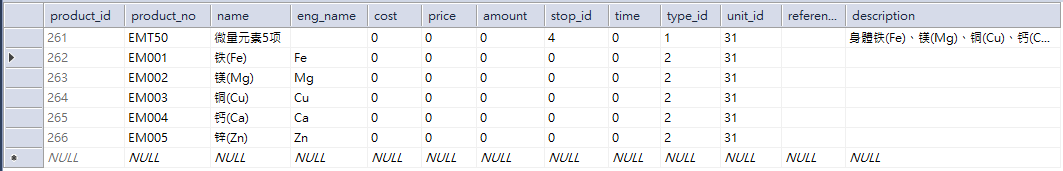
EM002 镁(Mg) eng\_name=Mg stop\_id = 0 type\_id = 2 unit\_id=31

EM003 铜(Cu) eng\_name=Cu stop\_id = 0 type\_id = 2 unit\_id=31

EM004 钙(Ca) eng\_name=Ca stop\_id = 0 type\_id = 2 unit\_id=31

EM005 锌(Zn) eng\_name=Zn stop\_id = 0 type\_id = 2 unit\_id=31

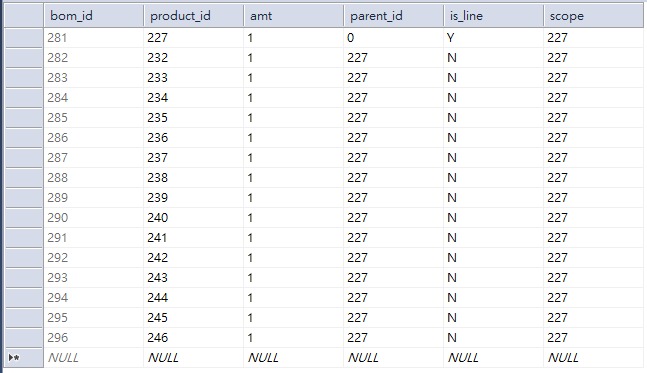
結果如下：



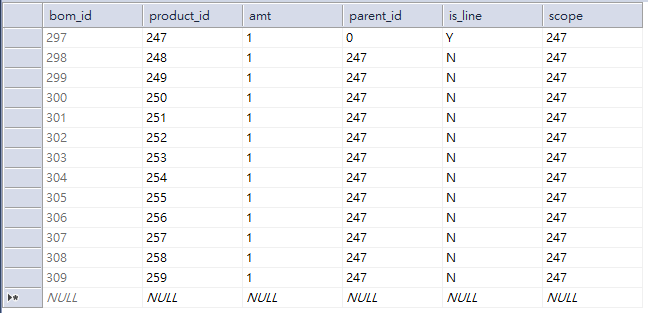
##### 新增 bom 資料(練習)

原始的 product 建置好之後，我們就要將這四個大項目加入 bom 表之中。為了增加速度，我們直接看結果圖。

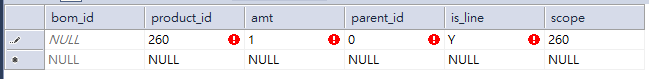
###### 內科檢查



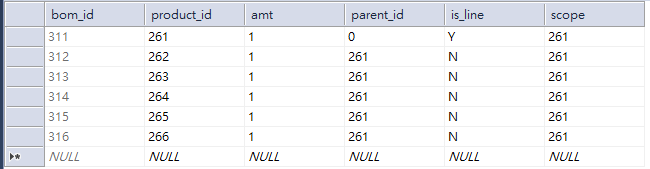
###### 耳鼻喉科檢查



###### 便常規

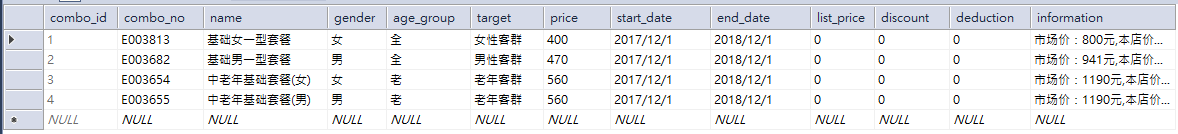


###### 微量元素5项



##### 新增 combo 資料(練習)

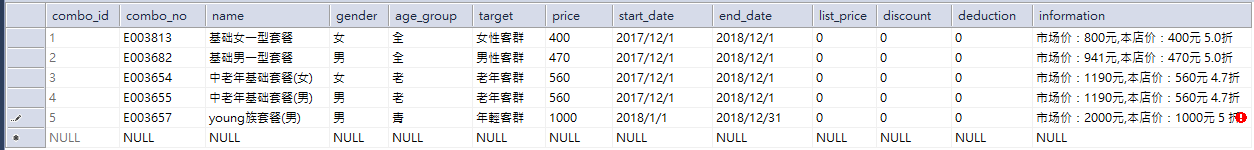
這個部份很簡單，就是建立一個套裝的名稱跟基本資料：



這些資料就是在客戶查詢套餐的時候會顯示出來的資料，介紹幾個重要欄位：

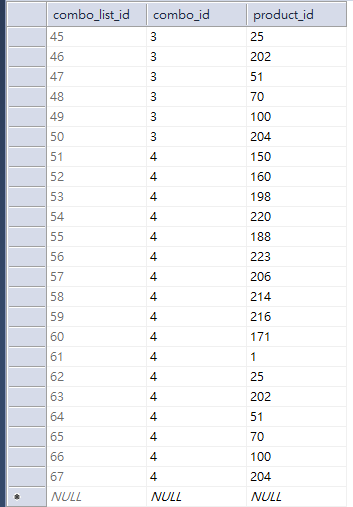
* Gender: 只會出現[男][女][全]，一般體檢男女的項目都不同，[全]為備用。
* Age\_group: 有可能出現[兒][青][成][老][全]分別代表兒童、青少年、成人、老人、全部適用。至於年齡的精確數值可另行定義。目前沒有訂死。
* Target: 目標族群，直接以文字表示。
* Information: 說明文字，會顯示在網頁上。

其他我們暫時不提。輸入結果如下：



##### 新增combo\_list 資料(練習)

最後一個步驟，就是把所有的檢查項目與套餐連接起來，透過combo\_list：



我們必須先知道所有項目的 product\_id，全部連接到 combo\_id = 5 上面。先將Excel表整理一下，只保留Level 1，如下圖：



然後，按照順序輸入即可：

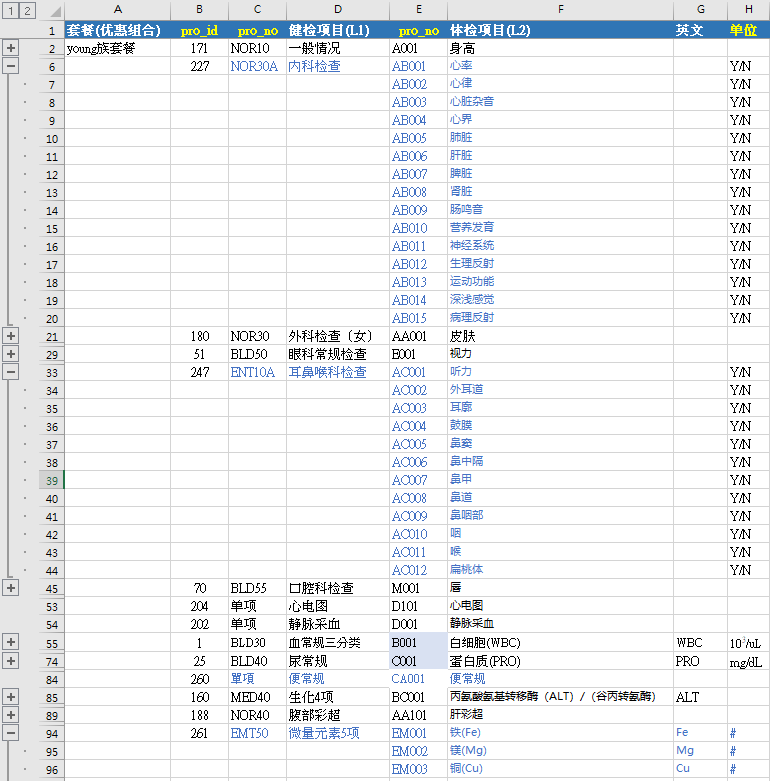


這樣就完成了一個套餐的輸入。

#### 三之六、完成 young 族套餐並修改combo欄位

在完成了 young族(男)體檢套餐之後，我們已經知道如何從我們的系統分析檔案 -- Excel 體檢表\_Michael.xlsx 裡面整理並且實際輸入一個套餐，必須說明，這個Excel 表格裡面所有的id都是我們的範本資料庫的內容，如果你自行建立的套餐與這裡不同，不用緊張，到時候只要導入範本資料庫就解決了。這樣做是為了保證每個人都在相同的資料庫底下開發。

我先按照上一個章節建立好的 product\_id，把 young族(女)整理如下：



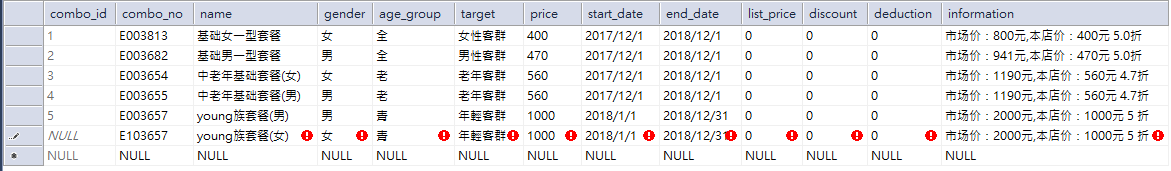
藍色的部分就是我們直接從young族(男)複製過來的，結果在這裡我們完全不需要建立新的 product ，連bom 表都不需要，只要直接建立 combo 與 combo\_list 就好。由此可知，當我們的 product 與 bom 建立的越完整，後面的速度就會越快。

參考下面兩張圖，我們直接來比較這兩個套餐的差異：





其實只有 “外科檢查” 不同而已。其他完全相同。因此我們直接打開 combo 資料表，將資料輸入如下：



為了方便輸入，可以剪貼以下文字：

E103657

young族套餐(女)

女

青

年輕客群

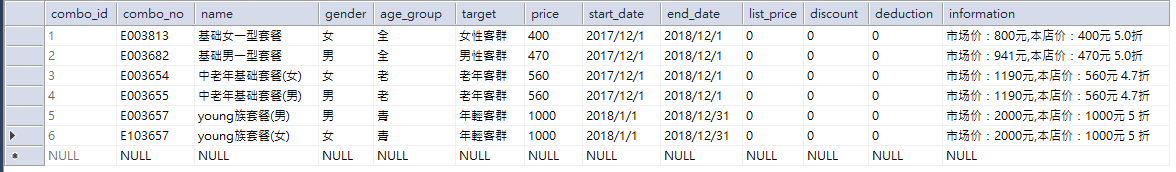
1000

2018/1/1

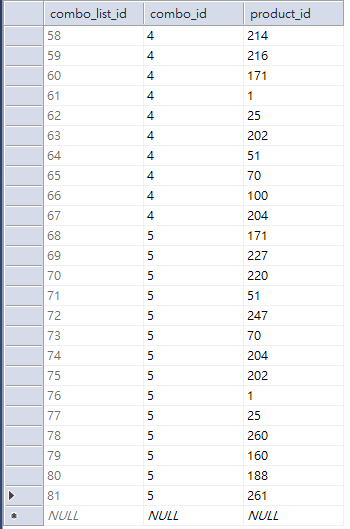
2018/12/31

市场价：2000元,本店价：1000元 5 折

輸入完成後自動產生combo\_id：



然後開啟 combo\_list，將資料游標移動到最後面，如下圖：



輸入 combo\_id = 6 的所有 product\_id (或許你的id跟我不同，請再檢查一次)：

171

227

180

51

247

70

204

202

1

25

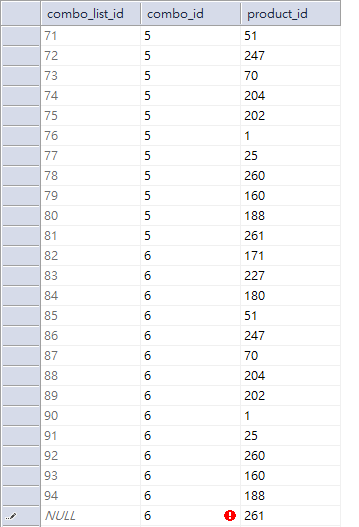
260

160

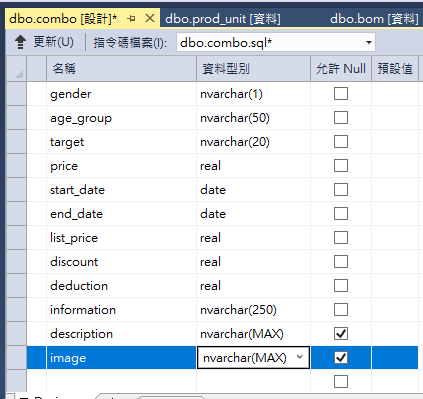
188

261

完成：



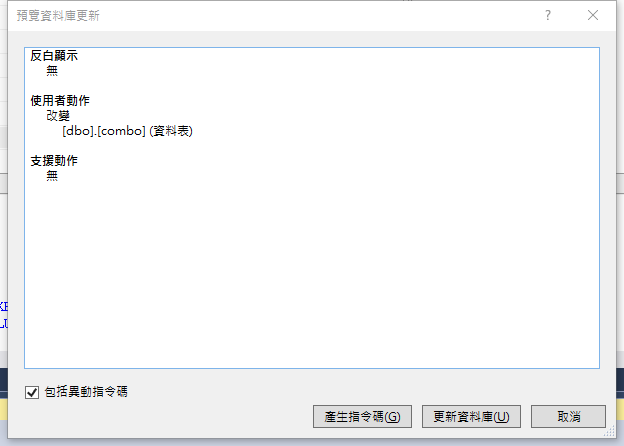
最後，由於 combo 資料表必須在網頁中動態顯示，我們需要多加兩個欄位來存放說明文字與圖片，如下圖：



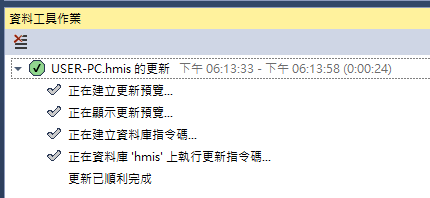
description nvarchar(MAX)

image nvarchar(MAX)

按下左上角的[更新]，出現下圖：



再按[更新資料庫]即可。



PS: 如果這時候出現無法修改，那是因為沒有開放權限，SQL Server 內定是不開放修改權限的，這時候就要進入 SQL Server Management Studio 裡面將權限打開。

##### 資料修改操作

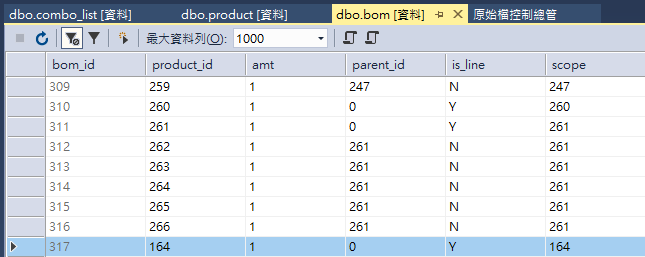
完成了上面的步驟之後，我們馬上進行兩種操作，第一是將一個 type\_id = 2 的小項目，變成客戶可以選擇的 “單項” 體檢項目。另一個是由 “單項” 轉變為 “組合” 的操作。

第一種操作的例子是BC004空腹血糖(FPG)，原本這是一個包含在生化四項裡面的小項目，如下圖：



但是因為這個項目很多客戶會單獨選擇，例如客戶選了血脂四項而沒有選生化四項，而血脂四項裡面沒有包含空腹血糖，所以希望可以再單獨選空腹血糖。

要讓BC004空腹血糖(FPG) 可以單獨被選取，只要加一行到 bom 表裡面就行，空腹血糖的 product\_id = 164，如下圖：



接這處理第二種情況，前面我們將 “便常規” 定義成了單項項目，但是後來我們發現便常規應該是一個組合項目，內容如下：大便性状、大便颜色、大便红细胞、大便白细胞、大便虫卵。當然，Excel 表裡面的內容，我們也做了更正，如下圖：



首先，按照這裡的更正，我們先找到在 product 資料表裡面原本的 “便常規”，product\_id = 260，將product\_no 改成 BLD60，這是因為編號原則的原因，原本的CA001將會被下面的小項目所使用。接著，從 CA001開始，將這五個小項目新增的項目輸入到 product 資料表裡面，依照下面的字串複製貼上：

CA001 大便性状 stop\_id = 4 type\_id = 2 unit\_id=35

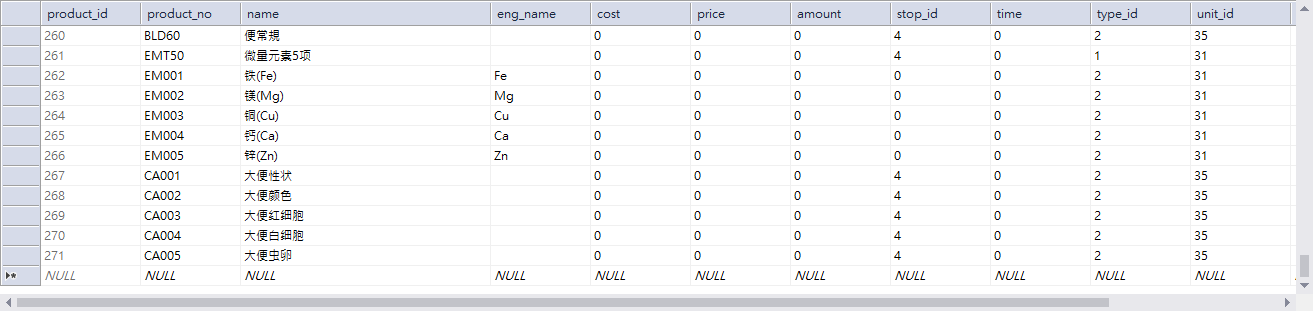
CA002 大便颜色 stop\_id = 4 type\_id = 2 unit\_id=35

CA003 大便红细胞 stop\_id = 4 type\_id = 2 unit\_id=35

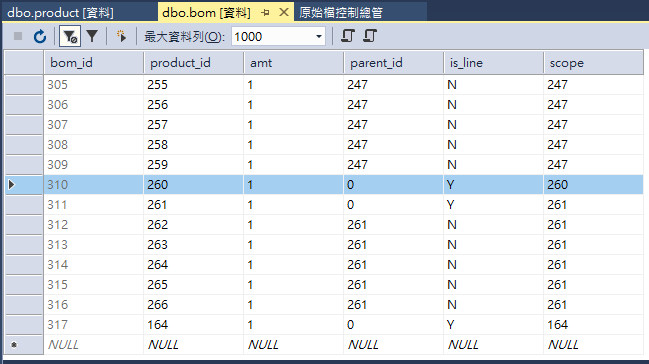
CA004 大便白细胞 stop\_id = 4 type\_id = 2 unit\_id=35

CA005 大便虫卵 stop\_id = 4 type\_id = 2 unit\_id=35

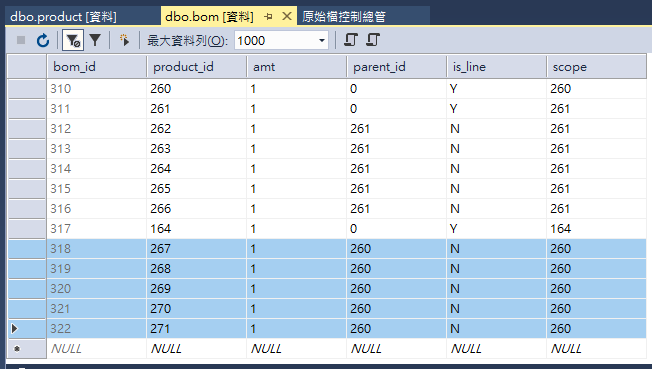
請記得，其他數字欄位填0，文字欄位刪除NULL。結果如下：



接著要來處理bom表，原本的bom 表如下圖：



我們可以看到 product\_id = 260 只有一行，不必在意下面是否已經有別的資料加入了，我們接著往下新增就行。新的便常規小項目的 product\_id = 267 到 271。加入後如下圖：

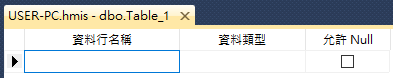


這樣就修改完畢了，不需要再去修改 combo 資料表了，因為 combo 裡面紀錄的是 product\_id = 260，與該項目是單項或是組合無關，然後由這些 product\_id 可以查到 type\_id，1是組合，2是單項。

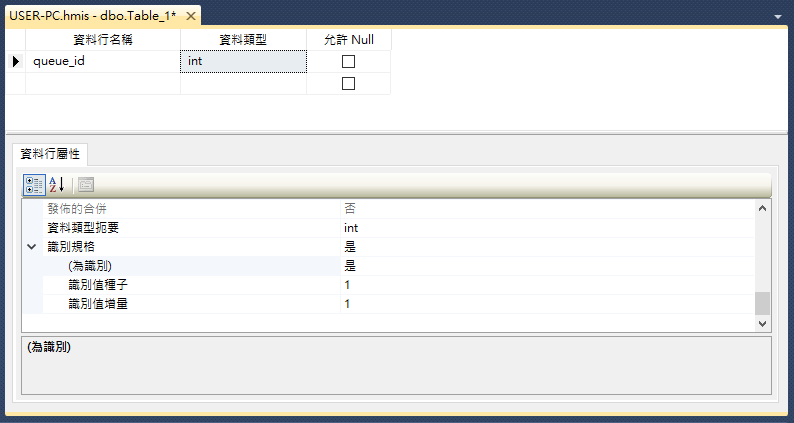
#### 三之七、新增排隊資料表

目前 Alex 的進度到了排隊功能，原本的資料庫已經無法使用，必須新增一個 queue 的資料表。

打開 SQL Server Management Studio 工具，連接Local 資料庫，打開 hmis 資料庫，點選[資料表]按下滑鼠右鍵，選取[新增][新增資料表]，出現下圖：



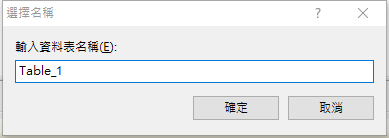
先輸入第一個欄位 queue\_id，由於是 primary key，所以比較特別，輸入名稱後，資料型態選 int，然後在下圖的[識別規格]中，選 “是”、 “1”、 “1” 即可，如下圖：



接著把其他欄位設定完畢：



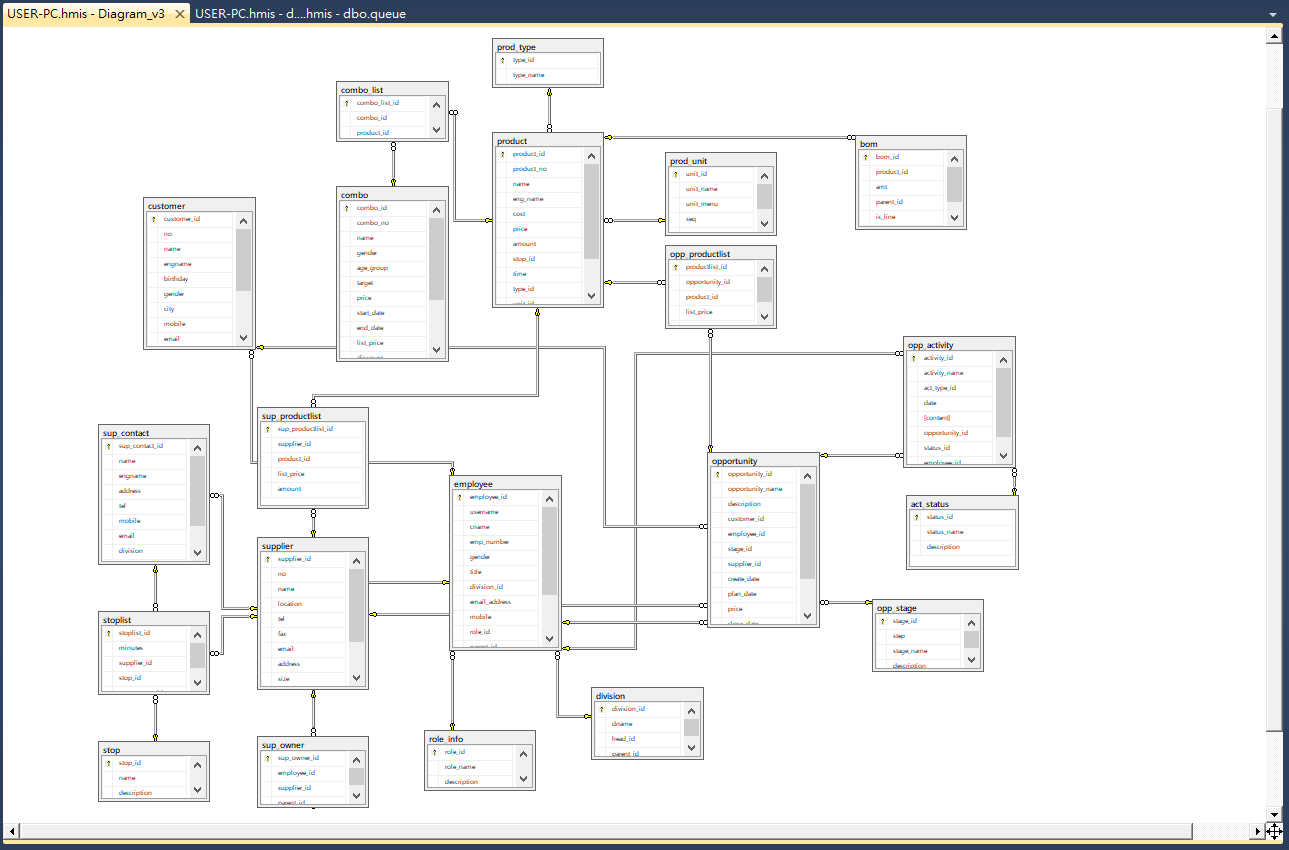
請確定所有欄位名稱與資料型態都正確無誤，就可以存檔了。按下儲存按鈕，就會出現下面的對話框：



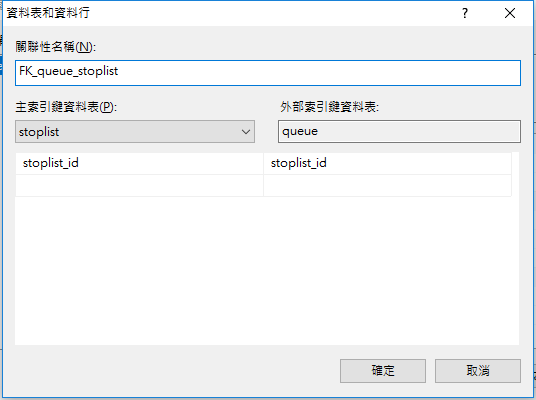
將 Table\_1 改為 queue 即可。

資料表新增完成之後，接下來就是關聯了，從新增的 queue 資料表的資料欄位名稱應該不難看出，其中 stoplist\_id, customer\_id 是外部鍵，我們必須修改資料庫圖表。

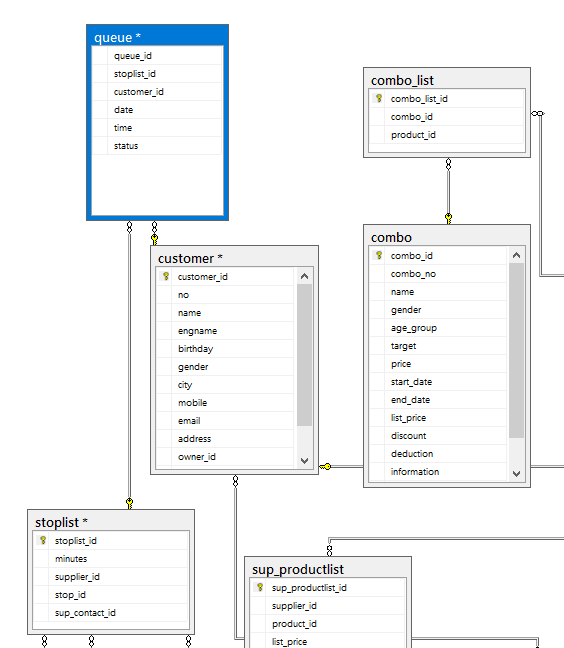
双擊已存在的 dbo.Diagram\_v3，可以看到下圖：



在畫面的空白處按下滑鼠右鍵選取[加入資料表]，然後選取 [queue]再按[加入]按鈕，新增的 queue 資料表就會出現，用拖曳的方式，將 stoplist 資料表裡面的 stoplist\_id 拖曳到 queue 資料表裡面的 stoplist\_id，就會出現下圖：



然後按下[確定]兩次，關聯就建立了，同樣的方式，customer\_id 也是如此方法，完成之後如下圖：



最後案儲存將此關聯表存起來，就大功告成。