Introduce twMVC

Who We Are and What We Want



為何組成 twMVC

- 我們熱愛 Web 開發技術，並樂於分享 ASP.NET MVC 開發心得
- 已成功建立數個灘頭堡的經驗，深知 MVC 開發模式條理分明，有利於分工與管理，想推薦給尚未了解的開發人員
- 不願意看到 MVC 叫好不叫座，我們樂意透過經驗交換來破除導入障礙

~~當然也希望能趁機紅起來~~

twMVC 資訊分享

- A quite simple http://mvc.tw/



- But, it’s still under-construction now ☺

twMVC 做哪些事推廣 ASP.NET MVC 技術

- 每週小型聚會，不特定主題

颱颱風千萬別來，~~我們沒那麼熱血~~

  - 固定於每週四 19:30-21:30，不改時間，逢國定假日暫停一次
  - 議程不會提前排定，以當天參加者最近的心得分享為主

- 舉辦不定期研討會

twMVC 不定期研討會

- 預先設定主題的中大型會議，主題提前於 mvc.tw 上公告
- 原則上一個月排定一次，初期因場地關係設定報名人數上限為 30 人

場場都是精心安排的實戰經驗

ASP.NET MVC網站的從無到有

從頭到尾完成一個MVC專案







主題

- 建立ASP.NET MVC專案
- 使用Razor與Entity Framework完成Model、View、Controller
- 邊做邊學ASP.NET MVC的特色

複習一下 什麼是 MVC

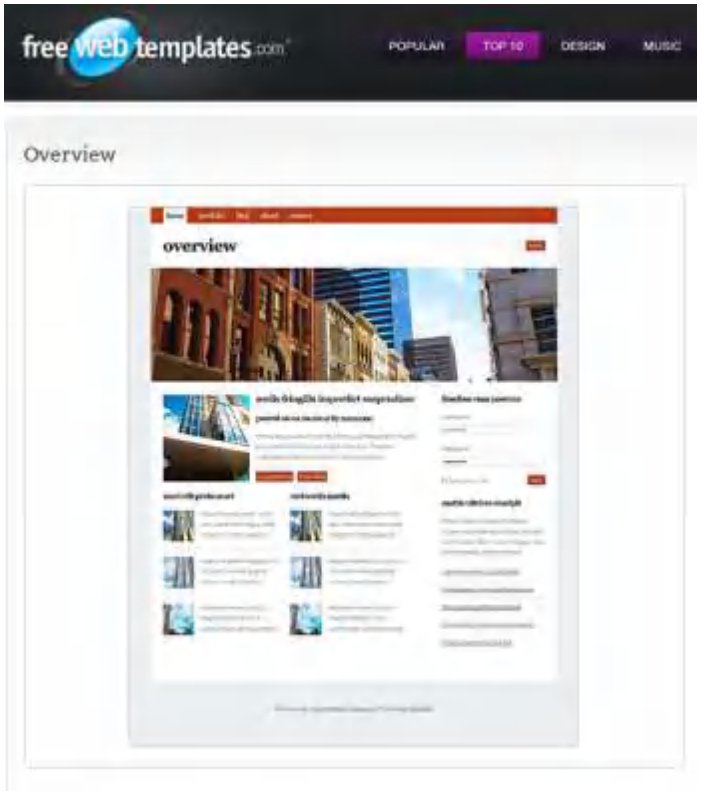
- 把應用程式架構給切分為三個部分，分別為：
  - Model（模型）
  - View（視圖）
  - Controller（控制器）
- MVC的三個組成部分所負責的功能不同，相互獨立
- MVC並不等於三層式架構

首先-開發環境的準備

- IDE
  - Visual Web Developer 2010 Express (Free)
  - Visual Studio 2010 (本次使用)
  - Visual Studio 2012 RC
- Visual Studio 開發套件
  - ASP.NET MVC 3 (本次使用)
  - ASP.NET MVC 4 RC
- 組件(佈署的主機上要安裝)
  - ASP.NET MVC 3 (本次使用)

專案的建立與套版

- 建立ASP.NET MVC 3 Web Application
- 使用免費版型

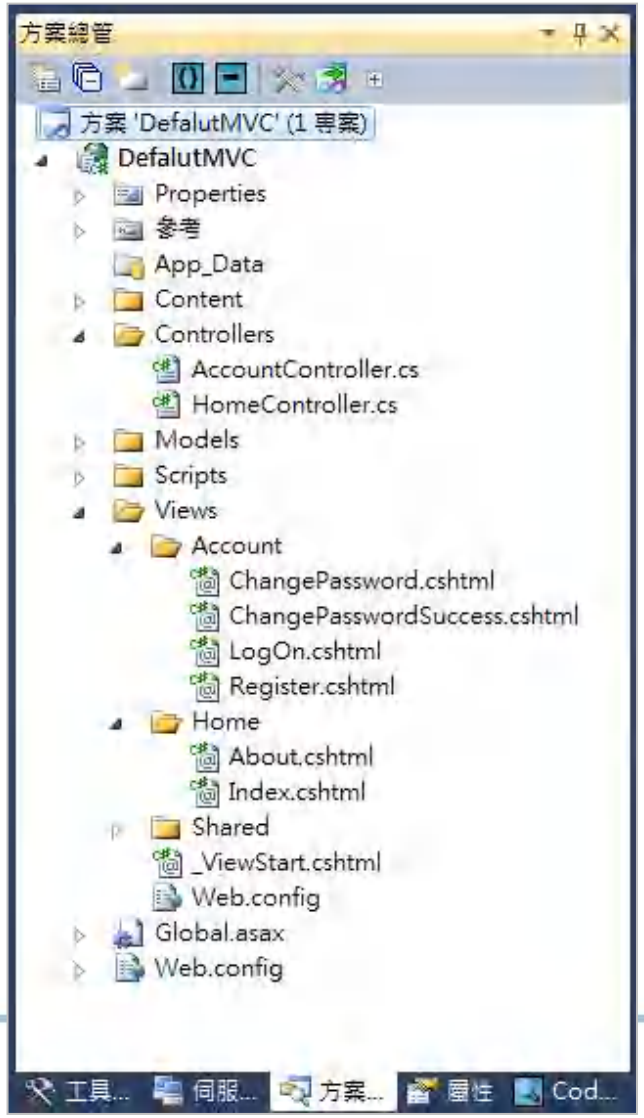


實際操作 1 開始MVC專案

1. 建立AspNetMvc.StartFromScratch專案
2. 複製檔案到Content中

認識ASP.NET MVC習慣的目錄結構

- App_Data
  - 內部使用的資料(無法使用URL存取)
- Content
  - 靜態檔案(CSS、Images)
- Controllers
- Models
- Scripts
  - JavaScript
- Views(無法使用URL存取)



實際操作 2 建立Layout與套用

1. 建立_LayoutBase.cshtml
2. 建立_LayoutMain.cshtml並套用_LayoutBase.cshtml
3. 使用UrlHelper更換Url
4. 建立HomeController
5. 建立Home/Index.cshtml並套用_LayoutMain.cshtml
6. 測試

建立Layout與Partial頁

- Layout與Partial主要是將每頁複雜的項目簡化，且都可以重覆使用。
- Layout又稱母頁或主頁，
  - 使View可以套用共同的設計，但一個View只套用一個Layout，但Layout可以有父Layout(巢狀Layout)。
- Partial
  - 是將View切成一個一個小部分，每個View可以載入任意的Partial。
- 延伸學習
  - ASP.NET MVC 3: Layouts and Sections with Razor

Razor語法

以 @ 開始

- @ = <% Response.Write(Html.Encode(code)) %>
- @() 同 @ 但可以寫比較複雜一點的Code(有空白，有運算式)
  - 如@(A==ture ? “Yes” : “No”)
- @{} 程式碼區塊
- @: 在 @{ @:直接輸出文字 }
- @* *@ 註解
- 延伸學習
  - ASP.NET MVC3 Razor 初心者容易遇到的問題

Razor Keyword

- @model
- @section
- @helper

- 延伸學習
  - ASP.NET MVC 3: New @model keyword in Razor
  - ASP.NET MVC 3 and the @helper syntax within Razor

Razor Method

- @RenderBody 呈現子View的主體
- @RenderSection 呈現子View @section{ }內容
- @RenderPage 呈現檔案內容
- @Write

- 延伸學習
  - ASP.NET MVC 3 Razor Syntax - RenderBody, RenderPage and RenderSection

UrlHelper

- 使用Url.Content或Url.Action產生Url的優缺點
- 優點
  - 以應用程式的根為起始點，不會因為部署的環境，而產生路徑問題
    - http://appdomain1/ < 原本部署
    - http://appdomain2/myAPP/ < 改成這裡沒處理好會有路徑問題
  - 根據Route產生Url
- 缺點
  - 會花一點效能

ViewData ViewBag TempData

- ViewData與ViewBag資料的存放是同一個，但呼叫方式不同
  - ViewData[“Key”]
  - ViewBag.Key
  - 生命週期為一個Action
- TempData如同TempData，但生命週期會一個Request
- 如果一個Request會有很多RenderAction請使用TempData。

實際操作 3 資料的讀取

1. 建立Entity Framework 4.0 DemoModel.edmx檔案
2. 從資料庫產生Entity
3. 建立BlogController
4. 建立Part Action
5. 使用Linq取得5筆Blog資料
6. 建立與完成Blog/Part.cshtml
7. 修改Home/Index.cshtml
8. 測試

建立Controller

- 依功能別建立Controller
- 如:
  - HomeController
  - BlogController
  - NewsController
  - AccountController

(P.S.建議不要全部都寫在HomeController中)

建立Model-使用 ORM

- 什麼是ORM

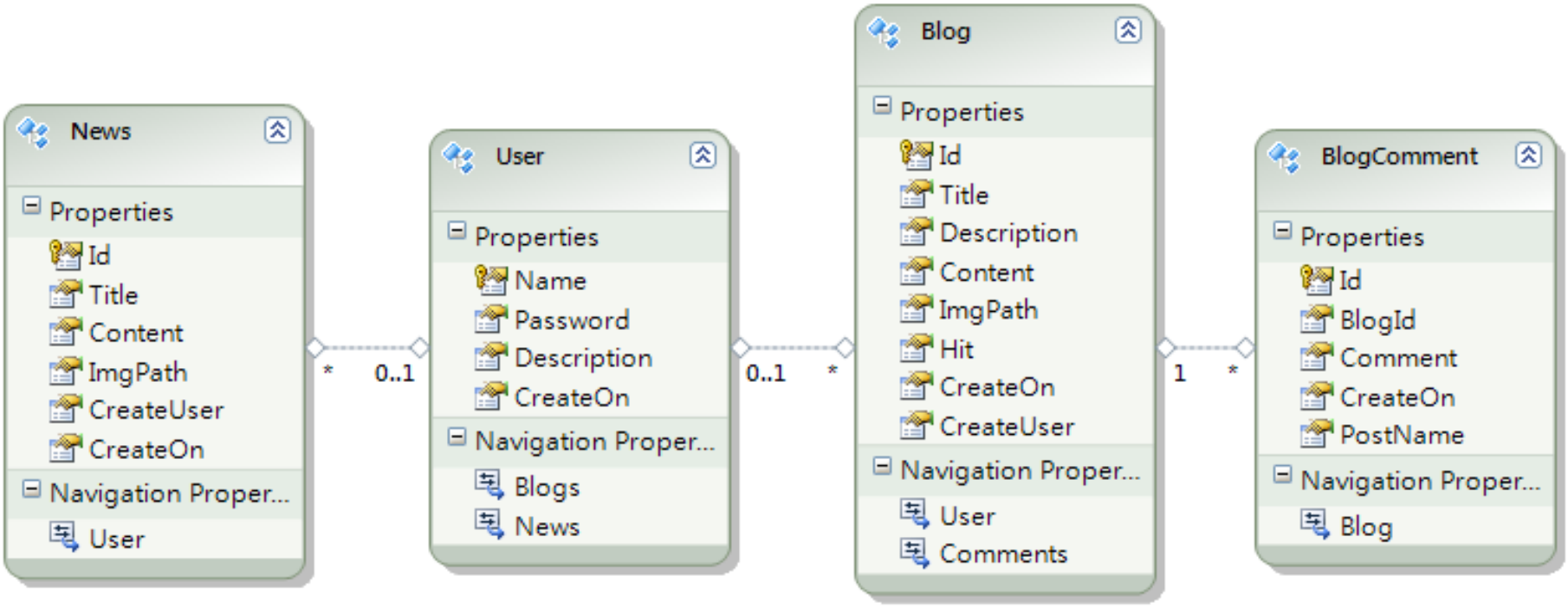
  物件關聯對應（Object Relational Mapping，簡稱ORM，或O/RM，或O/R mapping），是一種程式設計技術，用於實現物件導向編程對不同類型系統的數據之間的轉換。



ORM有什麼好處

- 在N-Tire構架中，撰寫商業邏輯層的人，可以專注在處理流程，不需要分心去思考如何存取Database。
- 可以快取切換Database。
  - 雖然正式環境很改變Database，但測試時很方便，如單元測試時使用SQLite。
- 開發速度快
  - 有強大Visual Studio IntelliSense與LINQ支援
  - 而且看Entity比看Table來的輕鬆
- 重覆使用
- 擴充性高

使用VS的EF工具產生Entity



實際操作 4-1 LazyLoad

- 建立Blog/Hot.cshtml

  使用LazyLoad取得CreateUser資料，發生

  ObjectContext 執行個體已被處置，無法再使用於需要連接的作業。

1. 在Global.asax.cs中增加CurrentEntities
2. 修改BlogController
3. 測試

ORM的Context處理 For EF

- 每一個EF的Entities都有Context(上下文)，處理Entity的狀態，用來管理物件是否有修改過、LazyLoad使用的Connection，與物件的快取等等。
- 如果網站只使用同一個Context，很容易因為多執行緒造成衝突，常見的作法是一個Request使用一個EF Context。

什麼Lazy-Load

- Lazy-Load是ORM中很重要的特色，會在使用關連時自動查尋與載入關連資料，如:

```
Blog blog = entities.Blogs.First(x=>id==id);
string name = blog.Comments[0].CreateUser.Name;
```

請問上二行程式自動下了幾次SQL

什麼Lazy-Load(2)

- 總共 ： 3 次 你猜對了嗎
- Lazy-Load自動化下Sql的時機是在Property的第一次使用的資料，Lazy-Load在完全不需要任何處理就可以取得關連表的資料是不是很方便。
- 但Lazy-Load也有不適用的地方，如上一張投影片的程式如在for(n<1000)的迴圈中，每個迴圈3個Sql，總共會下3000次，效能會非常差，這時還是用Join只下一次Sql會比較好。

LINQ與ORM的延伸學習

- Entity Framework 概觀 – MSDN
- Entity Framework 快速上手&學習資源整理 - The Will Will Web
- LINQ 的簡介 – MSDN
- 101 LINQ Samples C#- MSDN

實際操作 4-2 Count資料

1. 使用Linq Count資料

   @this.Model.Comments.Count

   但發現效能不好，請問為什麼效能不好??

2. 重寫Count計算

3. 發現要寫好長的Namespace，增加預設Namespace

ORM 是雙面刃 很方便但要小心

- ORM很方便，可以 . 阿 .想要的資料就出來的，比寫Sql Join來Join去或要寫一堆的Sql才能取得資料方便太多了，但要小心，每一個 . 可能就會下載海量資料回來。

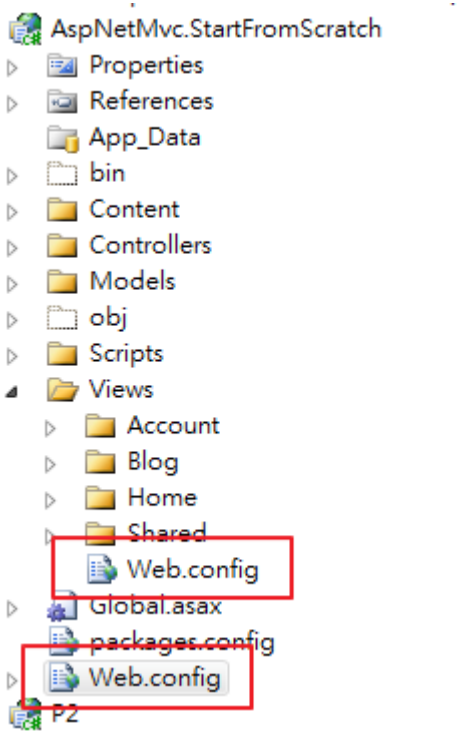
  如:

  Blog.Comments.Count，為什麼要將所有的Comment下載回來後才Count，為什麼不在資料庫端Count，只要下載一個int就可以了。

MVC專案的Web.Config為什麼有多個??

MVC有至少有二個Web.Config

- 根目錄下的Web.Config
  - 給應用程式使用
- Views下的Web.Config(Area也會有一個)
  - View的編譯設定
  - Request Block



- 廷伸學習

  [ASP.NET MVC 3 使用 Razor 如何在 Web.config 宣告預設的 Namespaces]

實際操作 5-1 AJAX下載BlogComment

1. 資料NuGet下載 jQuery
2. 新增BlogComments的Comments與AddComment的Action與View
3. 新增Demo.js
4. 使用 .Load() AJAX下載資料
5. 測試

使用NuGet

- NuGet是專案的套件管理，安裝簡單、自動判斷更新、刪除也不複雜。
  - 常見的jQuery、Fancybox、Elmah、NHibernate、Spring.Net都可以一鍵安裝，非常的方便。
- 因為是專門給Visual Studio用的會使用微軟習慣，如:
  - 安裝js會將檔案放在/Scripts/，圖檔會放在/Content/下
  - Dll會自動加入參考
  - 有設定檔會自動設定(如Elmah會修改Web.Config)
- 延伸學習
  - 還在揮汗徒手安裝程式庫? 試試NuGet吧 - 黑暗執行緒

什麼是jQuery

- 最熱門的JavaScript函式庫
  - 跨瀏覽器的DOM元素選擇
  - DOM巡訪與更改
  - 事件（Events）
  - CSS操作
  - 特效和動畫（移動顯示位置、淡入、淡出）
  - Ajax
  - 延伸性（Extensibility）
  - JavaScript外掛程式 等等等
- 廷伸學習

  jQuery - 维基百科

JavaScript路徑處理

- 因為現在現在網站沒有Ajax的很少，不可避免會在js檔中寫下後端的路徑，建議不要寫絕對路徑，以免更換部署方式，讓AJAX全死。
- 在Element上設定路徑

```
<a id="ajaxComment" href="/Wade/Blog/Comments/4"> 7 Comments</a>
```

```
$("#ajaxCommentPanel").load(this.href, function () {
```

- 在_Layout.cshtml中偷偷設定

```
    <input id="ApplicationPath" value="@Request.ApplicationPath" type="hidden"  />
</body>
</html>
```

```
> window.option.AppPath
  "/Wade"
> window.option.AppPath + "/Blog/AddComment"
```

實際操作 5-2

1. 在_LayoutMain.cshtml中設定應用程式路徑

   這裡使用藏在Script中的方式

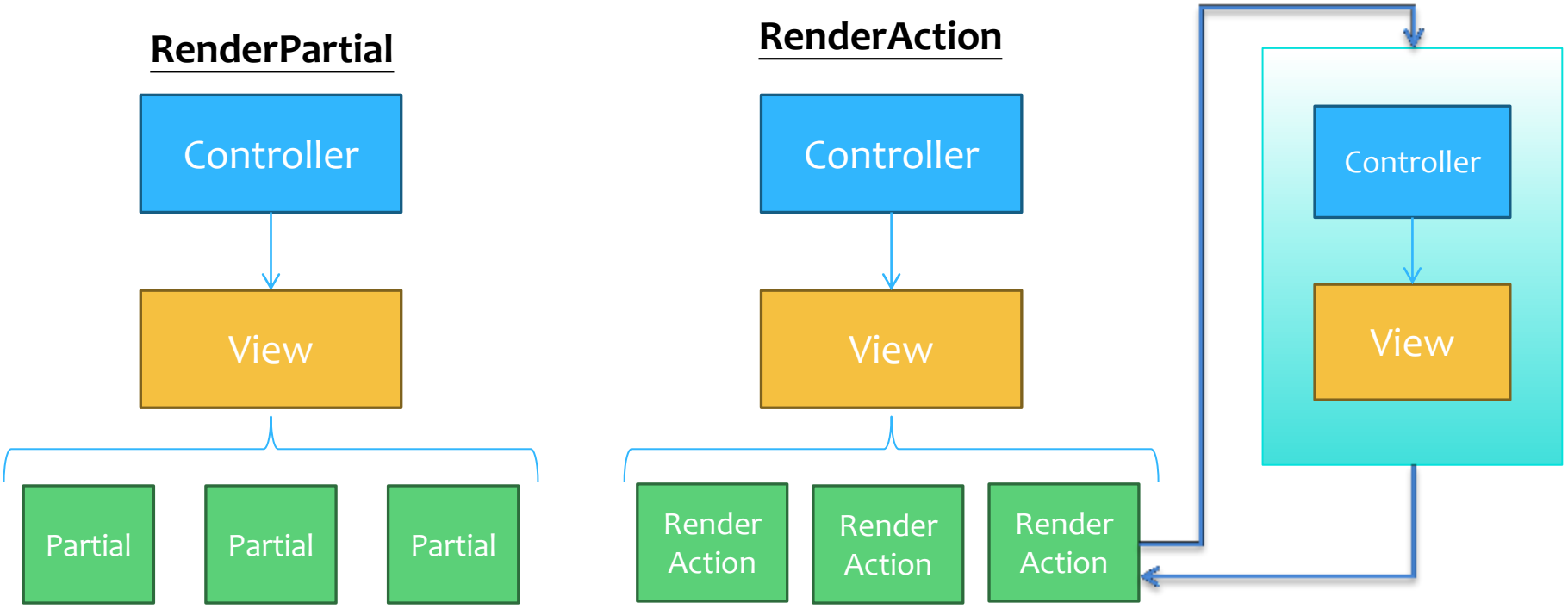
```
<script src="/Wade/Scripts/Demo.js" type="text/javascript">{"AppPath":"/Wade"}</script>
```

2. 更換部署方式

3. 測試AJAX是否正常

RenderPartial & RenderAction

- RenderPartial 與 RenderAction 的差異?



實際操作 6-1 Validation

1. 增加BlogComment的Metadata
2. 增加所需的js檔
3. 因為是AJAX下載資料，要重新綁定事件
4. 執行

Validation

- 使用System.ComponentModel.DataAnnotations可以輕鬆的做到前後端的驗證。
- 整合jQuery.validate，擴充性不會太複雜
- 常見的Validation有
  - Required、StringLength、RegularExpression、Range、Compare
- 延伸學習:
  - ASP.NET MVC3 如何使用內建驗證功能達到前端與後端的同時驗證 - demo小鋪

ModelBinding

- 會將Post或Get的值轉換 物件 與 使用HtmlHelper或AjaxHepler可以物件 轉換成 HTML元素。
- 同時會整個前後端Validation。
- 同時也會保存Post資料，可以還原輸入失敗的值。

```
@Html.TextBoxFor(m => m.UserName)
@* 這二個式子是相同的 *@
<input id="UserName" name="UserName" type="text"
       data-val="true" data-val-required="User name 欄位是必要項。"
       value="@(Request.Form["UserName"]==null ?
              (Model==null? null : Model.UserName) : Request.Form["UserName"])">
```

- 延伸學習:
  - ASP.NET MVC Model Binding - Part1 – CodeProject
  - ASP.NET MVC Model Binding - Part2 - CodeProject

HtmlHelper與AjaxHelper

- HtmlHelper與AjaxHelper，都是用來快速產生HTML的類別，同時整合ModelBinding與Validation，這些特色使ASP.NET MVC在資料的CRUD上，非常的輕鬆。

- 延伸學習:
  - ASP.NET MVC HTML Helpers - w3schools

實際操作 6-2 AJAX POST

1. 修改AddComment.cshtml，改用AjaxHelper
2. 增加Ajax OnComplete處理，重算Comment Count
3. 在Demo.js增加addCommentComplated function使用jQuery來計算Count。

RouteData

/Blog/Article/4

- 因為套用Route後Id不會在QueryString中，而是路徑，不能用Request[“Id”]取資料，需要改從RouteData中取資料。

```
this.ViewContext.RouteData.Values["id"].ToString()
```

實際操作 7

1. 建立LoginViewModel.cs
2. 建立AccountController
3. 建立Shared/_LogOnPartial.cshtml判斷是否登入
4. 建立Account/Login.cshtml
5. 修改Web.Config套用表單驗證

ViewModel

- ViewModel通常是為與View溝通用的，通常一個View不會剛好使用一個Table資料，或是剛好的欄位(如多處理分頁或排序用的欄位)

表單驗證與授權

- 表單驗證(FormsAuthentication)，ASP.NET 2.0之後內建的身份驗證模組，通過驗證後會用加密過的Cookie來身份。
- 整合HttpContent.Current.User。
- 可以使用web.conifg的authorization區段或AuthorizeAttribute做權限控制。
- 延伸學習
  - 概略解釋Forms Authentication的運作-The Will Will Web

更換Layout

- Layout(母頁)，不是寫死在View最開頭後就不能再改的，是可以更動的。
- MVC ViewEngine是先執行子頁，再依子頁中母頁設定，執行母頁，再將RenderBoay的地方換成子頁的內容，所以在子頁前都有機會更換Layout。

VirtualPathFactory View的搜尋

- View有一套預設的搜尋機制,使用相對路徑會，
  - 先找自己同一層資料夾下的
  - 然後在找Shared資料夾
- 絕對路徑使用”~/Views/”開頭。
- 使用絕對路徑要加上副檔名，不會自動判斷。

實際操作 8 CRUD

1. 建立Blog Add的Action
2. 撰寫EF Create的Code
3. 使用Create的Template建立View
4. 瀏覽時Update(Hit+1)
5. 執行

實際操作 9 自訂Route

1. Global.asax.cs在增加MapRoute
2. 測試

實際操作 10 部署

1. 設定Web.Release.Config
2. 在專案上按右鍵發行
3. 在IIS中建立新站台

部署

- 千萬千萬不用複製原始檔的方式佈署，請使用Visual Studio的發行功能。
- IIS 6要設定 ISAPI
- 需要安裝 ASP.NET MVC 3組件

- 延伸學習:
  - Razor Views 預編譯(Pre-Compile)-黃偉榮的學習筆記
  - ASP.NET MVC 3網站BIN目錄手動部署-黑暗執行緒
  - Web.config Minification 與 Transformation-小朱®的技術隨手寫
  - Visual Studio 2010 單鍵發行簡單使用 Web.Release.config - demo小鋪

課後作業

- 請來賓們回家完成News部份的操作



The End

謝謝各位