



# ASP.NET MVC 5 開發實戰

認識 Visual Studio 2013 開發環境



多奇數位創意有限公司

技術總監 黃保翕 (Will 保哥)

部落格：<http://blog.miniasp.com/>



All services from your imperative.

工欲善其事，必先利其器

# VISUAL STUDIO 2013 開發環境介紹



# 最近使用的專案可任意釘選或移除

All services from your imperative.

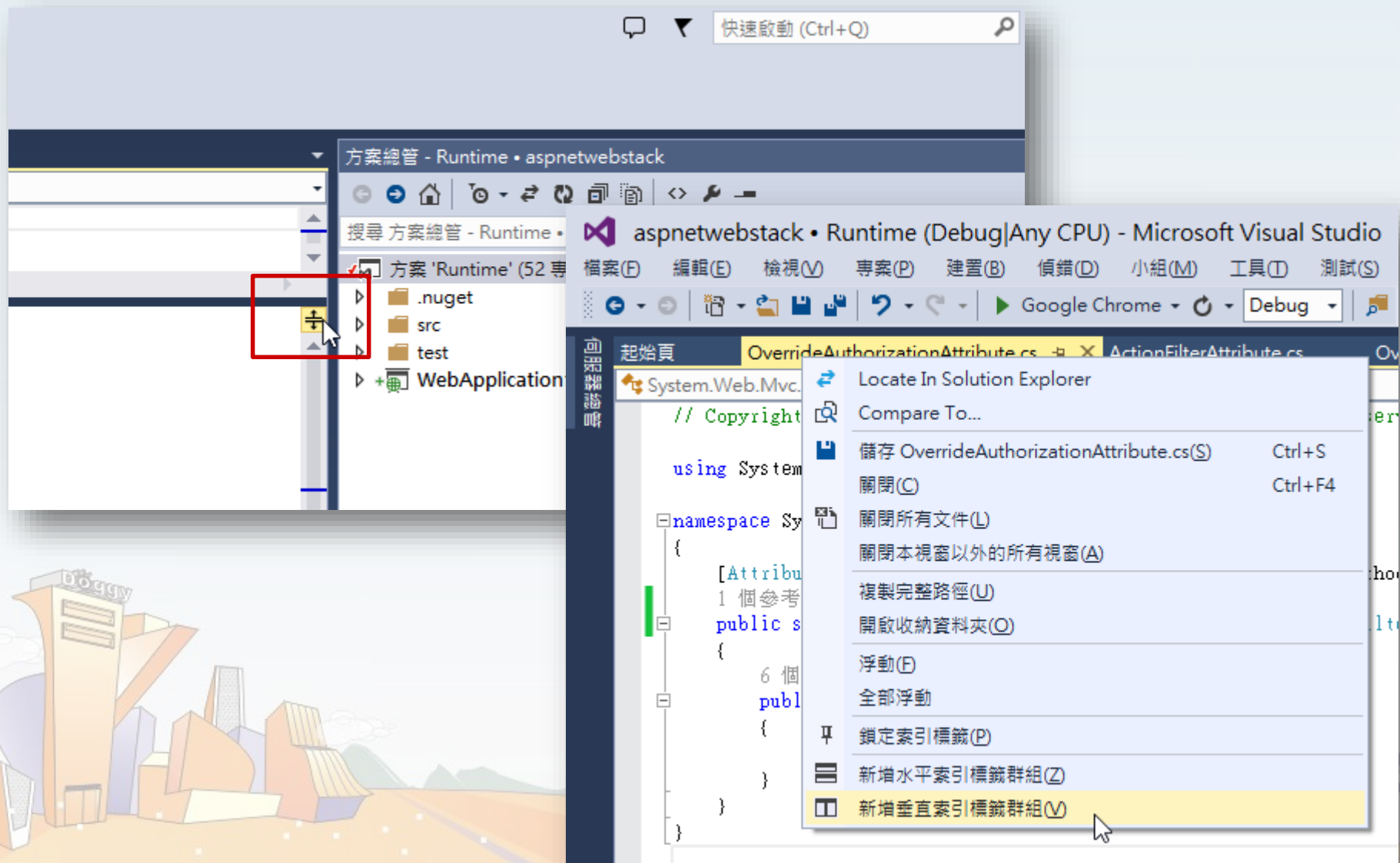
## □ 最近使用的專案

- 可自由釘選或移除
- 可開啟收納資料夾

The screenshot displays the Visual Studio 2013 Ultimate interface. On the left, a sidebar contains a '最近' (Recent) section with a list of projects. The first project, 'WebApplication1', is selected, and a context menu is open over it, showing three options: '開啟專案' (Open Project), '開啟收納資料夾' (Open Solution Folder), and '從清單移除' (Remove from List). The '開啟收納資料夾' option is highlighted. To the right of the sidebar, the main area is titled 'Ultimate 2013' and '產品影片' (Product Videos). Below the title, there is a section '利用以下短片，進一步了解' (Use the following short videos to learn more). This section contains four video thumbnails with titles and durations: 'Visual Studio 2013 Ultimate: 雲端負載測試' (1:53), 'Visual Studio 2013 Ultimate: CodeLens' (3:03), 'Visual Studio 2013 Ultimate: Managed 記憶體分析' (2:16), and 'Visual Studio 2013 Ultimate: 查看定義' (2:27).

# 分割視窗、字體縮放、停駐技巧

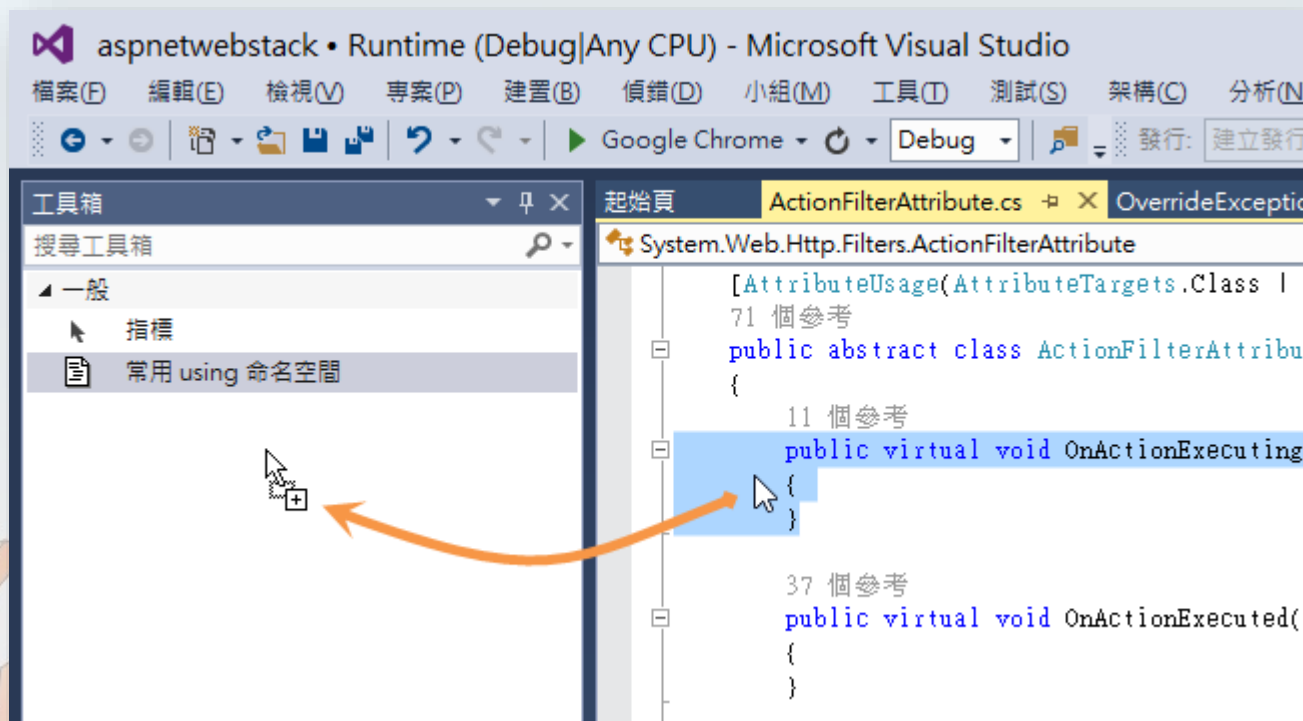
All services from your imperative.



# 活用工具箱

All services from your imperative.

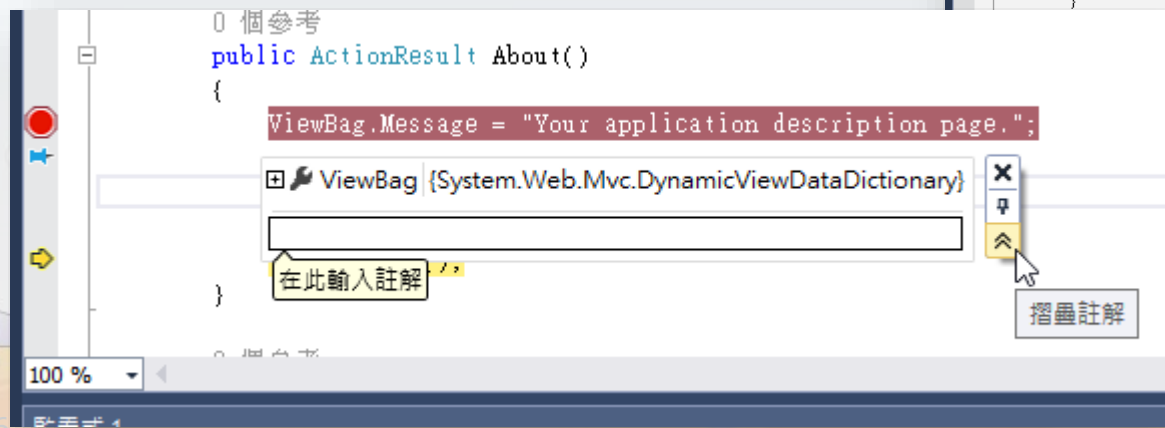
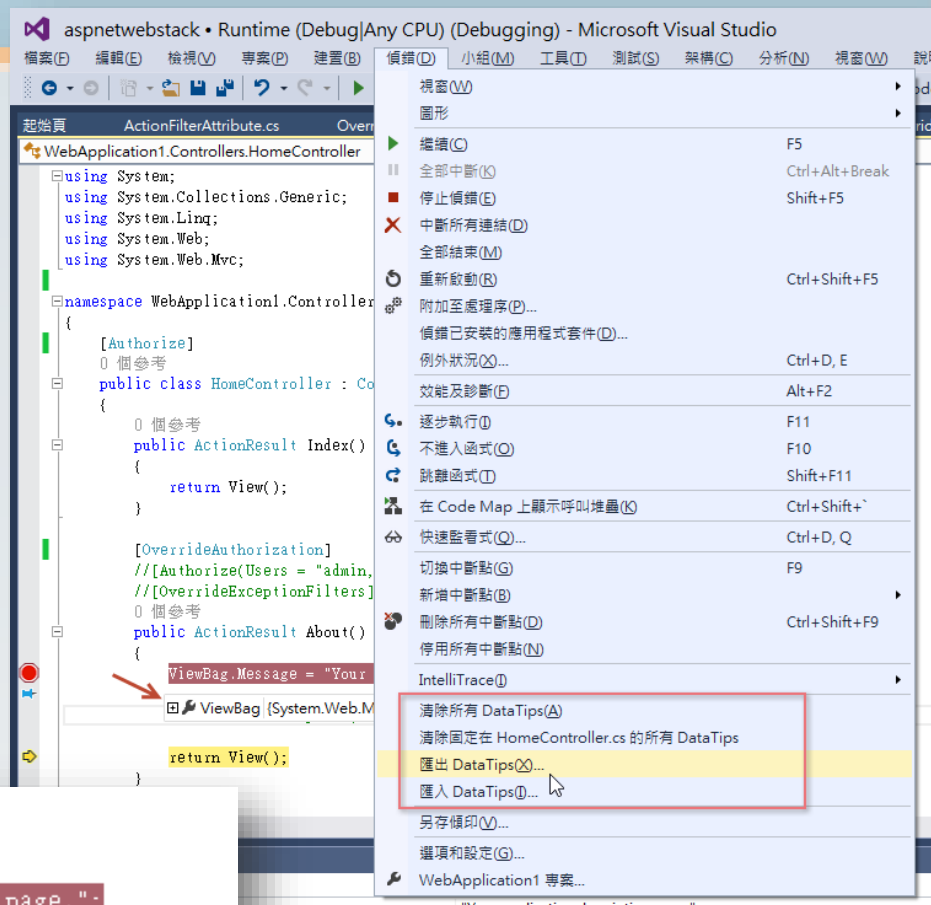
- 當成「剪貼簿」或「程式碼片段」來用
- 剪貼簿可重新命名項目，方便後續使用時參考



# 可浮動/可釘選的 DataTips

All services from your imperative.

- ❑ 可浮動 (絕對定位)
- ❑ 可釘選 (相對定位)
- ❑ 可匯入/匯出
- ❑ 可加上註解



# 伺服器總管

All services from your imperative.

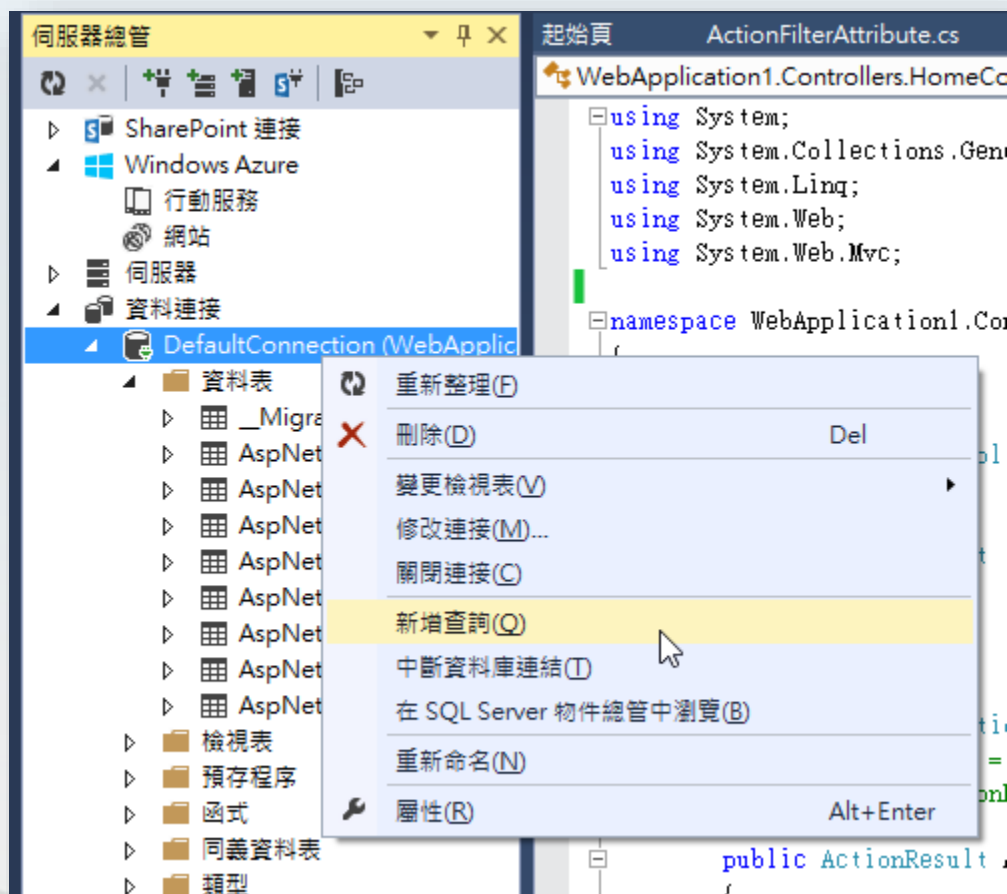
## □ 可從資料連線的角度管理

### ■ 資料連接

- 加入資料連接
- 建立新的資料庫

### ■ 特定資料庫連線

- 查詢連接字串
  - F4 開啟屬性視窗
- 變更檢視表
- 修改連接
- 關閉連接
- ---
- 新增查詢
- 在 SQL Server 物件總管中瀏覽





# SQL Server 物件總管

All services from your imperative.

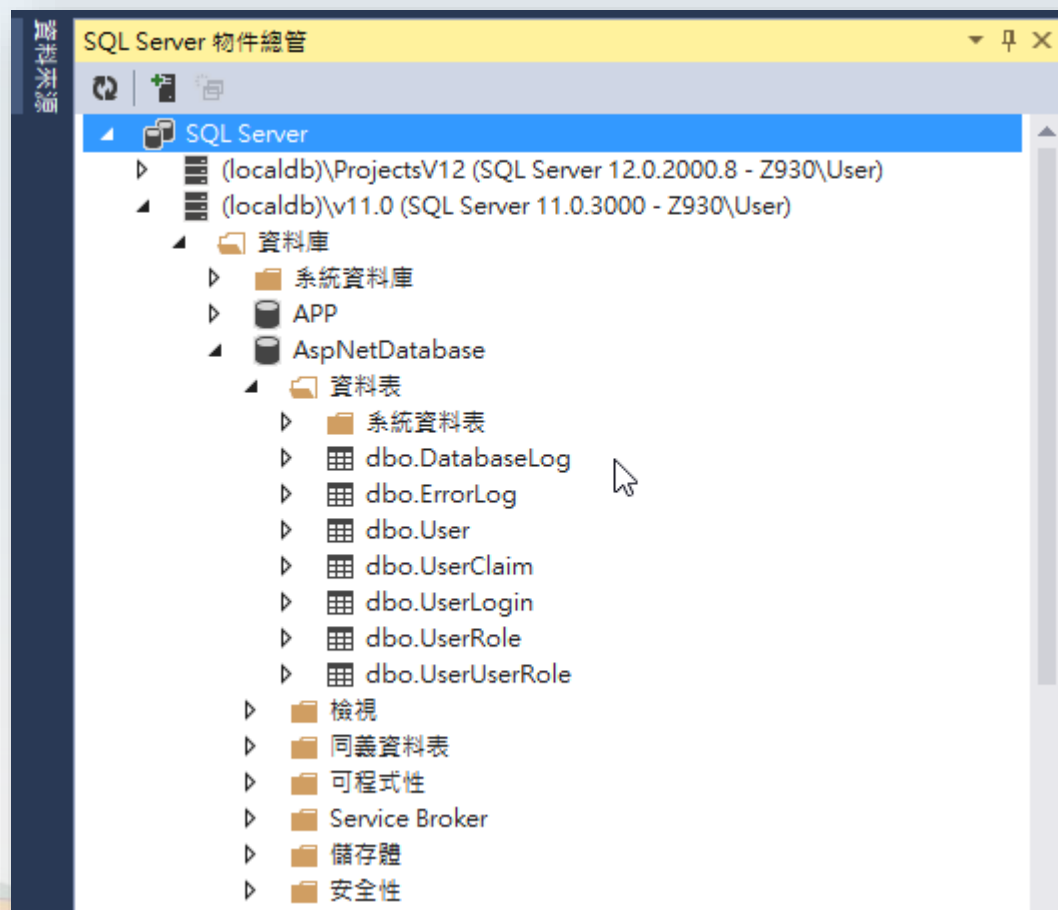
## □ 可從資料庫實體的角度管理

### ■ 特定資料庫

- 建立新專案
- 結構描述比較
- 資料比較
- -----
- 發行資料層應用程式
- 擷取資料層應用程式
- 註冊資料層應用程式
- 取消資料層應用程式

### ■ 特定資料表

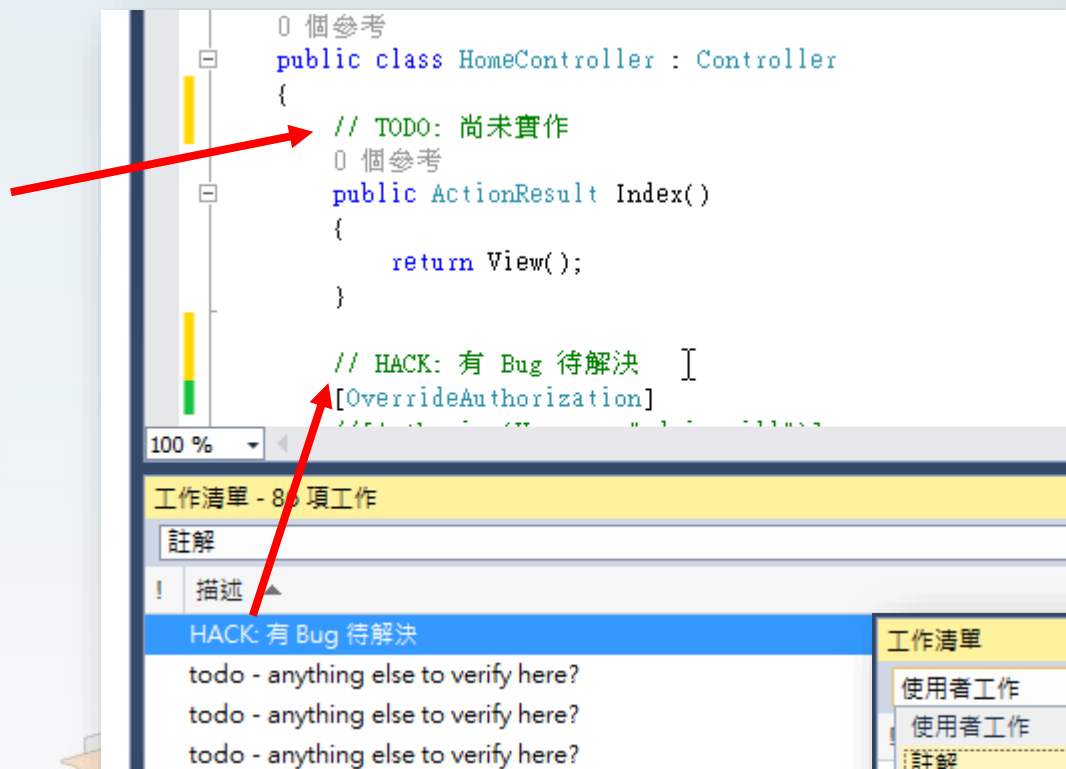
- 編寫指令碼
- 檢視表設計工具
- 檢視資料





# 工作清單

All services from your imperative.



0 個參考

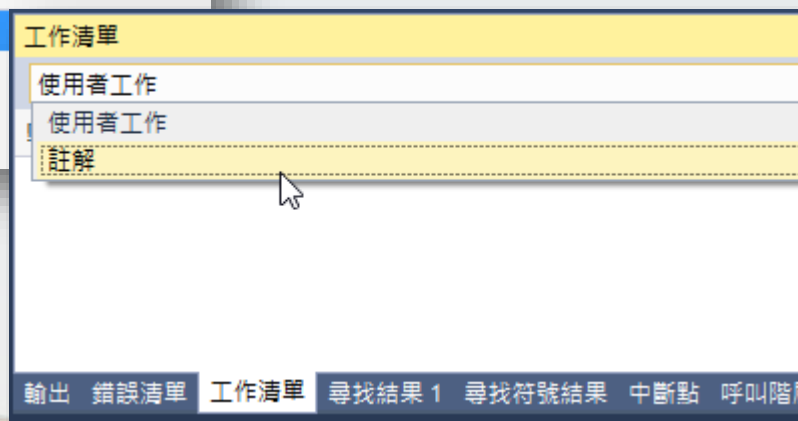
```
public class HomeController : Controller
{
    // TODO: 尚未實作
    0 個參考
    public ActionResult Index()
    {
        return View();
    }

    // HACK: 有 Bug 待解決 [
    [OverrideAuthorization]
    // ...
}
```

100 %

工作清單 - 8 項工作

註解
HACK: 有 Bug 待解決
todo - anything else to verify here?
todo - anything else to verify here?
todo - anything else to verify here?



工作清單

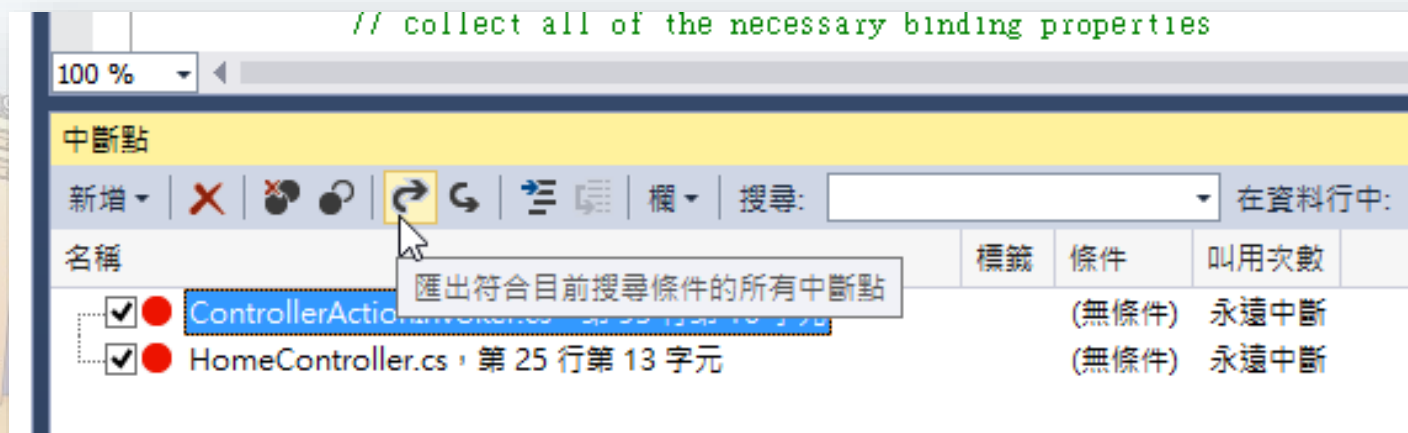
使用者工作
使用者工作
註解

輸出 錯誤清單 工作清單 尋找結果 1 尋找符號結果 中斷點 呼叫階層

# 中斷點強化管理機制

All services from your imperative.

- ❑ 將看到的中斷點匯出成 XML 檔案
- ❑ 從 XML 檔案匯入中斷點
- ❑ 搜尋所有中斷點 (可見的文字都可以搜尋)
- ❑ 可以讓你依據不同的模組進行標籤設定
- ❑ 叫用次數：可以讓你看出特定中斷點在本次偵錯中到底被呼叫了幾次。



All services from your imperative.

不可不知的操作密技

# VISUAL STUDIO 2013 操作技巧



# 增強的區塊選取與編輯功能

All services from your imperative.

- ❑ 可一次編輯多行 (區塊選取功能)
- ❑ 支援零寬度選取，可多行同時輸入
- ❑ 快速鍵
  - Alt + Shift + 鍵盤方向鍵
  - Alt + 滑鼠左鍵選取

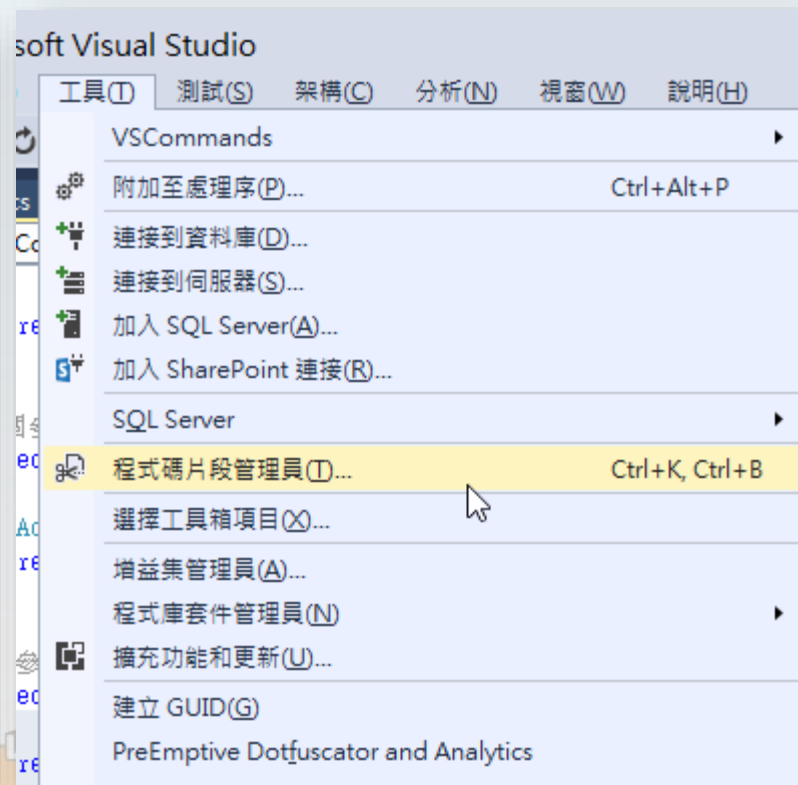
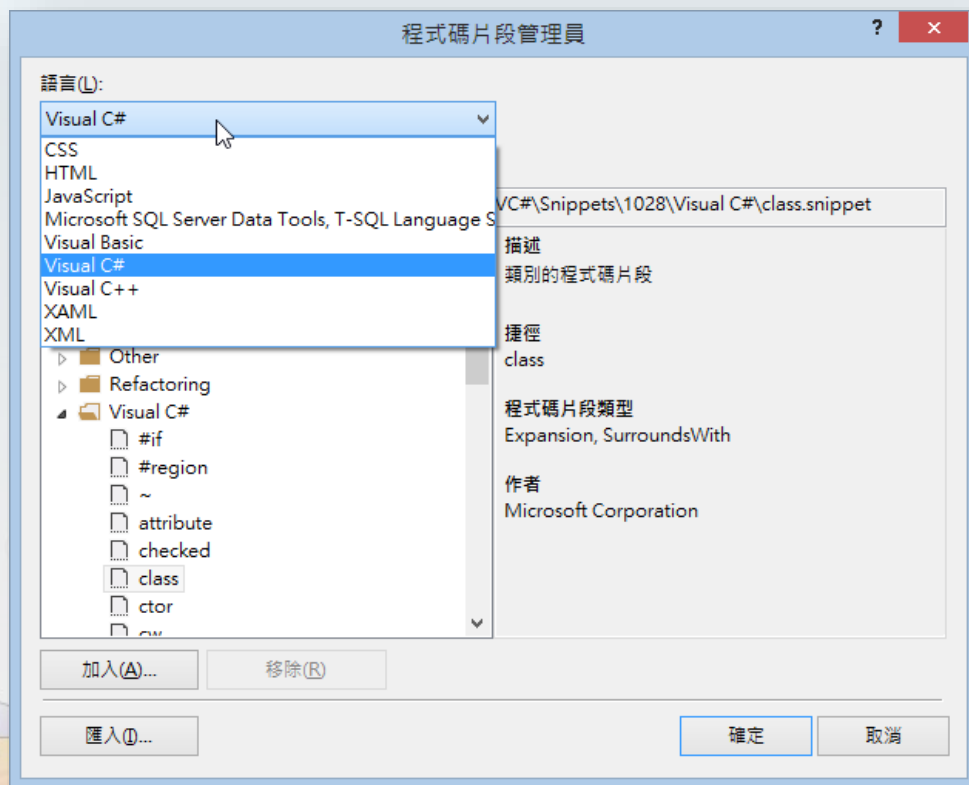
```
6 namespace ConsoleApplication1
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             Console.Wirte
13             Console.Wirte
14             Console.Wirte
15             Console.Wirte
16             Console.Wirte
17             Console.Wirte
18             Console.Wirte
19             Console.Wirte
20             Console.Wirte
21             Console.Wirte
22             Console.Wirte
23             Console.Wirte
24             Console.Wirte
25             Console.Wirte
26         }
27     }
28 }
```

# 程式碼片段管理員 ( Ctrl+K, Ctrl+B )

All services from your imperative.

## □ 支援的程式語言

- CSS 、 HTML 、 JavaScript 、 T-SQL Language 、 Visual Basic 、 Visual C# 、 Visual C++ 、 XAML 、 XML



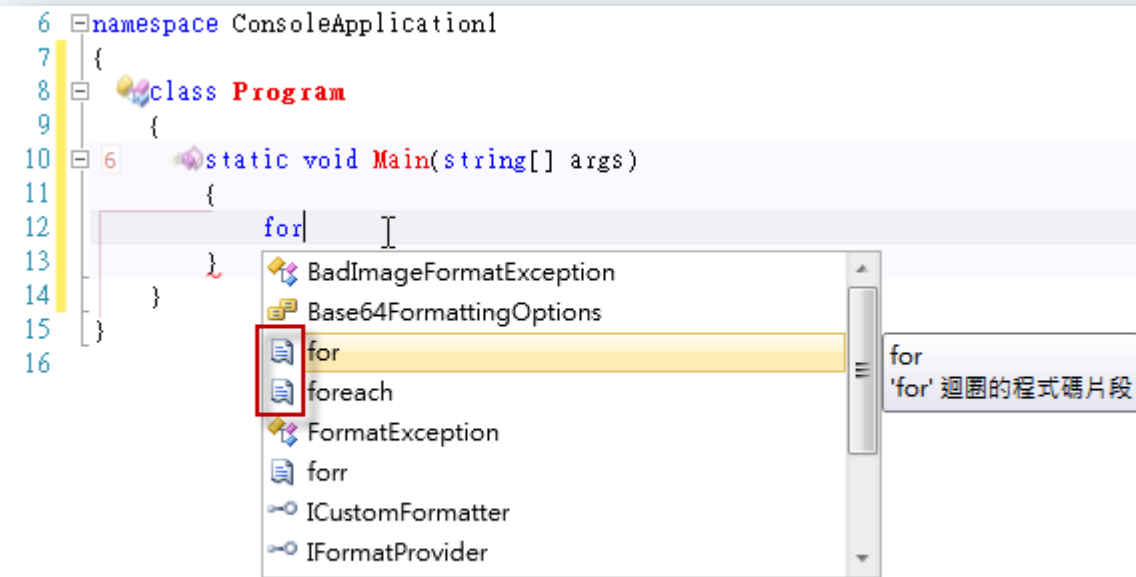
# 活用內建的程式碼片段 ( Visual C# )

All services from your imperative.

- ❑ if 、 else 、 for 、 forr 、 foreach 、 do 、 while
- ❑ namespace 、 class 、 interface 、 struct 、 enum
- ❑ ctor 、 using 、 prop 、 **propfull**
- ❑ try 、 tryf
- ❑ **switch**

```
var a = ConsoleColor.Black;

switch (a)
{
    case ConsoleColor.Black:
        break;
    case ConsoleColor.Blue:
        break;
    case ConsoleColor.Cyan:
        break;
    case ConsoleColor.DarkBlue:
```



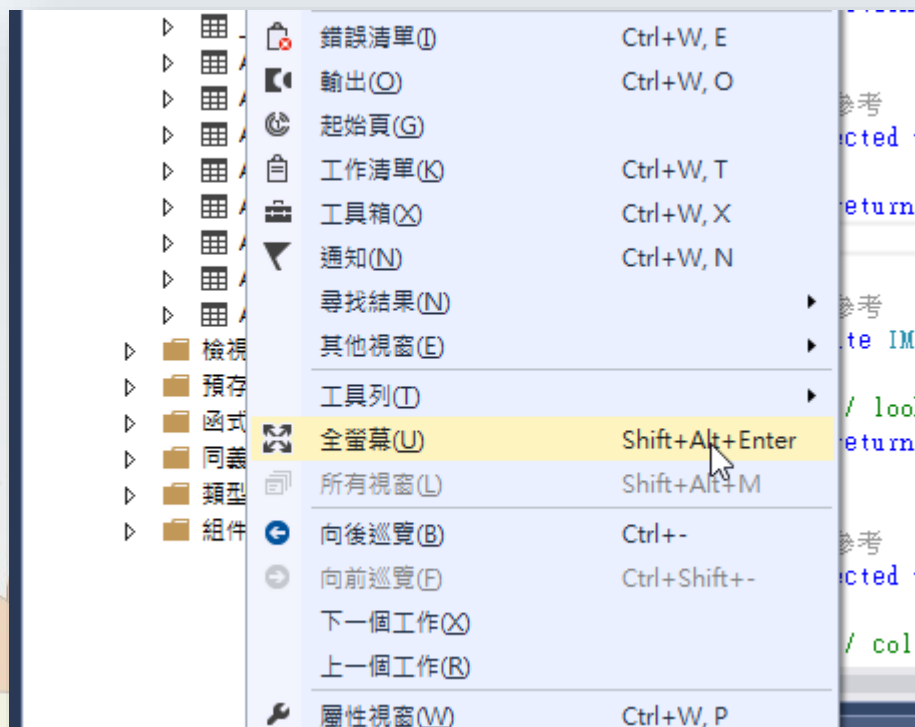
# 切換為全螢幕模式

All services from your imperative.

## □ 鍵盤快速鍵

■ Alt + Shift + Enter

## □ 檢視 → 全螢幕





# 必學的快速鍵 (1)

All services from your imperative.

## ☐ 選擇正確的鍵盤對應配置

### ■ Visual C# 2005

## ☐ 偵錯時常用

### ■ F6

建置方案 (Build Solution) ( Ctrl + Shift + B )

### ■ F9

切換中斷點

### ■ F10

單步執行 ( 不進入函式 )

### ■ F11

單步執行 ( 進入函式 )

## ☐ 搜尋快速鍵

### ■ Ctrl + E

將鍵盤游標移至搜尋框 (檔案總管也能用)

## ☐ Intellisense 相關

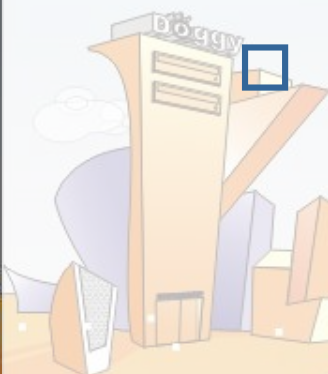
### ■ Ctrl + j

開啟 Intellisense 選單

☐ Visual Studio 2010 之後的版本都支援 **PASCAL CASE** 比對 (要輸入大寫字母)

### ■ Alt + →

自動完成 Intellisense 名稱



# 必學的快速鍵 (2)

All services from your imperative.

## □ 智慧標籤

- Ctrl + . (小數點) 開啟智慧標籤 (Smart tag)
- 智慧標籤種類
  - 引用命名空間
    - 找不到怎麼辦？先載入參考組件！
  - 實做 Interface 方法

## □ 瀏覽/搜尋程式碼

- F12 移至定義
- Ctrl + - (減號) 向後巡覽
- Ctrl + , (逗號) 巡覽至
  - 可先停留在特定關鍵字上，並按下 **Ctrl + ,** 直接搜尋整個方案
  - Visual Studio 2010 以後皆支援 PASCAL CASE 比對
- Shift + F12 尋找符號結果
  - 需先將鍵盤游標移至變數上再進行搜尋所有用到的地方

# 必學的快速鍵 (3)

All services from your imperative.

## ☐ 你不知道的複製、剪下、貼上

- Ctrl + C

不選取複製整行

- Ctrl + X

不選取剪下整行

- Ctrl + V

直接貼上整行

## ☐ 大綱模式

- Ctrl + M + L

切換所有大綱

- Ctrl + M + M

切換大綱展開 (最近一個大綱)

- Ctrl + M + O

摺疊至定義

## ☐ 重構

- F2

重新命名變數、方法、類別

## ☐ ASP.NET MVC

- Ctrl + M + G

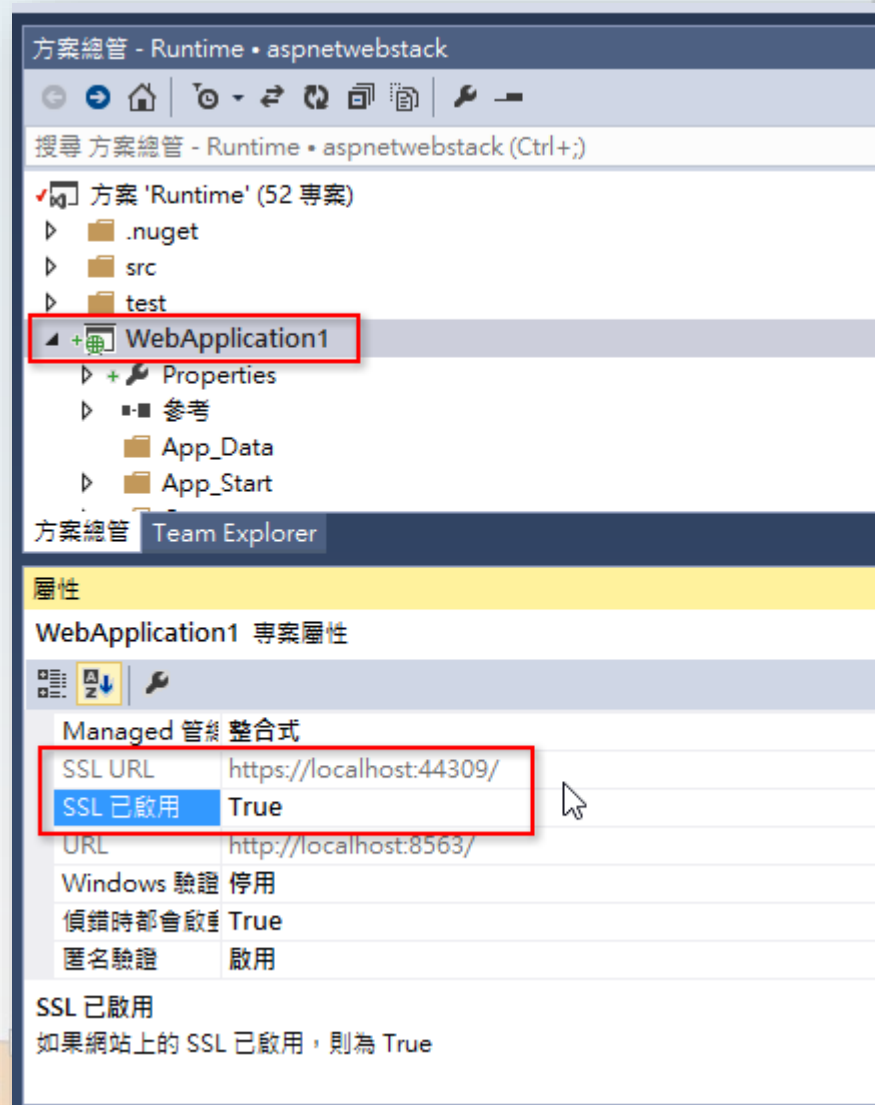
切換於 View 與 Controller 之間



# 啟用 SSL 加密連線網站

All services from your imperative.

- ☐ 專案節點屬性
  - SSL 已啟用: True
- ☐ 專案屬性
  - Web -> 起始動作
    - ☐ 指定頁
  - Web -> 伺服器
    - ☐ 專案 URL:  
請貼上 SSL URL
    - ☐ 建立虛擬目錄



方案總管 - Runtime • aspnetwebstack

搜尋 方案總管 - Runtime • aspnetwebstack (Ctrl+;)

- ✓ 方案 'Runtime' (52 專案)
  - ▷ .nuget
  - ▷ src
  - ▷ test
  - ▶ **WebApplication1**
    - ▷ Properties
    - ▷ 參考
      - App\_Data
      - App\_Start

方案總管 Team Explorer

屬性

WebApplication1 專案屬性

Managed 管線 整合式

SSL URL	https://localhost:44309/
SSL 已啟用	True
URL	http://localhost:8563/
Windows 驗證	停用
偵錯時都會啟用	True
匿名驗證	啟用

SSL 已啟用  
如果網站上的 SSL 已啟用，則為 True

# 瀏覽器連結: Browser Link

All services from your imperative.

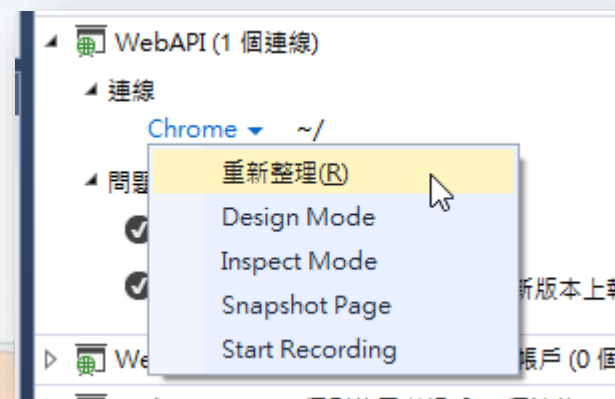
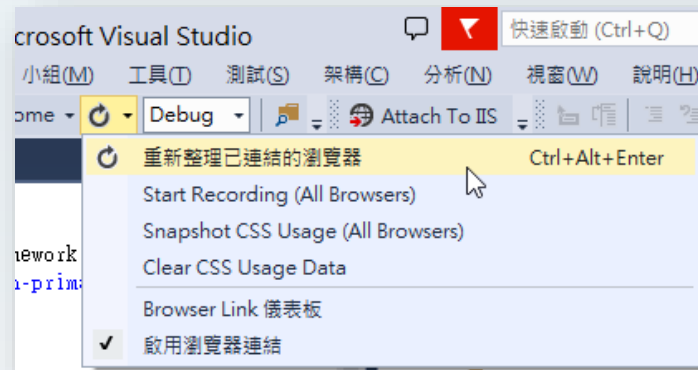
## □ Browser Link 工具列

- 重新整理已連結的瀏覽器( Ctrl+Alt+Enter )
- Browser Link 儀表板
- 啟用瀏覽器連結

## □ Browser Link 儀表板

- 重新整理

## □ 可透過 APIs 擴充功能



# Web Essentials 2013

All services from your imperative.

## ☐ Browser Link 擴充功能

<http://vswebessentials.com/features/browserlink>

- Auto-hide 預設隱藏選單，可按 Ctrl 開啟選單
- Save F12 changes 將 F12 修改的內容回寫 CSS 檔案
- F12 auto-sync 啟用 F12 修改自動同步 CSS 功能
  - ☐ 先修改 App\_Start\BundleConfig.cs 檔案
    - BundleTable.EnableOptimizations = false;
  - ☐ 只能自動回寫自訂的 CSS 檔案 ( bootstrap.css 不能自動修改 )
- Best practices 自動提供網頁優化建議
  - ☐ Web Developer Checklist
    - <http://webdevchecklist.com/>
  - Find Unused CSS 自動找出頁面中未使用的樣式

# Web Essentials 2013

All services from your imperative.

## □ Browser Link 工具列

- Start Recording (All Browsers) 錄製 CSS 使用狀況 (跨網頁)
- Clear CSS Usage Data 清空 CSS 使用狀況的歷史紀錄
- Enable Saving F12 Changes 啟用 F12 變更自動儲存功能

## □ Browser Link 儀表板

- Design Mode 驚豔的設計模式!
- Inspect Mode 自動找出網頁對應的 View 檔案
- Start Recording 開始錄製該連線的 CSS 使用狀況





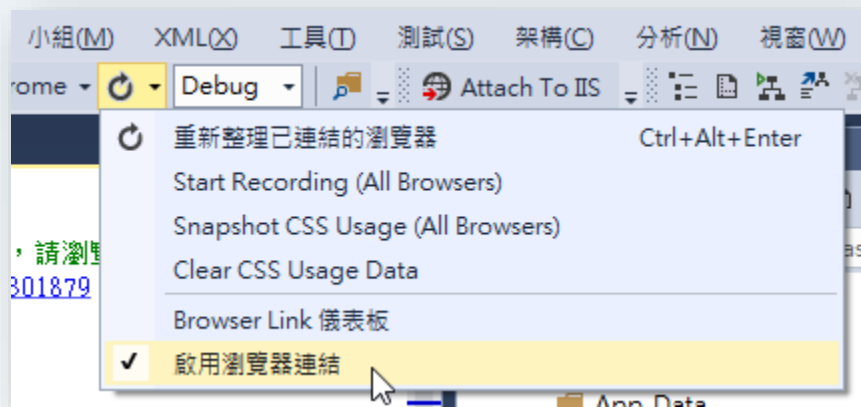
# 啟用或關閉瀏覽器連結

All services from your imperative.

## ☐ 設定 web.config 啟用偵錯模式 (必要)

- `<compilation debug="true">`

## ☐ 透過 Browser Link 工具列



## ☐ 透過 web.config 設定 appSettings

- `<add key="vs:EnableBrowserLink" value="false"/>`

# 靜態 HTML 的瀏覽器連結

All services from your imperative.

## □ 調整 web.config 設定 (適用 IIS7 以上版本)

```
<configuration>
  <system.webServer>
    <handlers>
      <add name="Browser Link for HTML" path="*.html" verb="*"
        type="System.Web.StaticFileHandler, System.Web,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        resourceType="File" preCondition="integratedMode" />
    </handlers>
  </system.webServer>
</configuration>
```

## □ 相關連結

- Using Browser Link in Visual Studio 2013  
<http://bit.ly/using-browser-link>

All services from your imperative.

不可不知的擴充套件

# VISUAL STUDIO 2013 擴充功能和更新



# Visual Studio 2013 擴充功能和更新主要功能

All services from your imperative.

## □ 安裝、移除、自動更新 擴充套件

### ■ \*.vsi

- 巨集、增益集、程式碼片段和特定其他 Visual Studio 擴充類型

### ■ \*.vsix

- 專案範本、項目範本、工作箱項目、Managed Extension Framework (MEF) 元件和 VSPackage

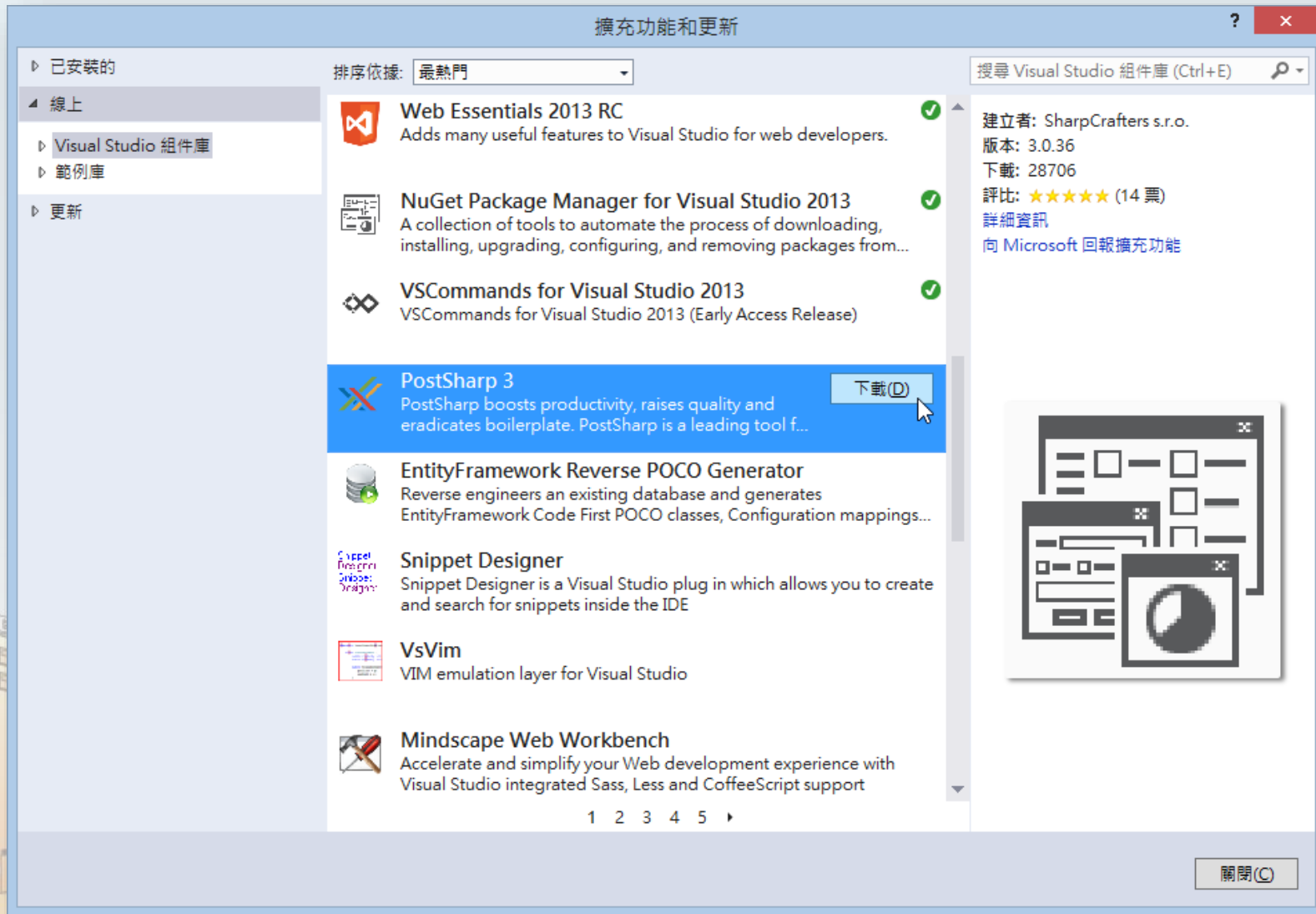
### ■ \*.msi

- 此類型擴充無法從 Visual Studio 2010 擴充管理員自動安裝，但可以在擴充管理員內瀏覽



# Visual Studio 2013 擴充功能和更新

All services from your imperative.



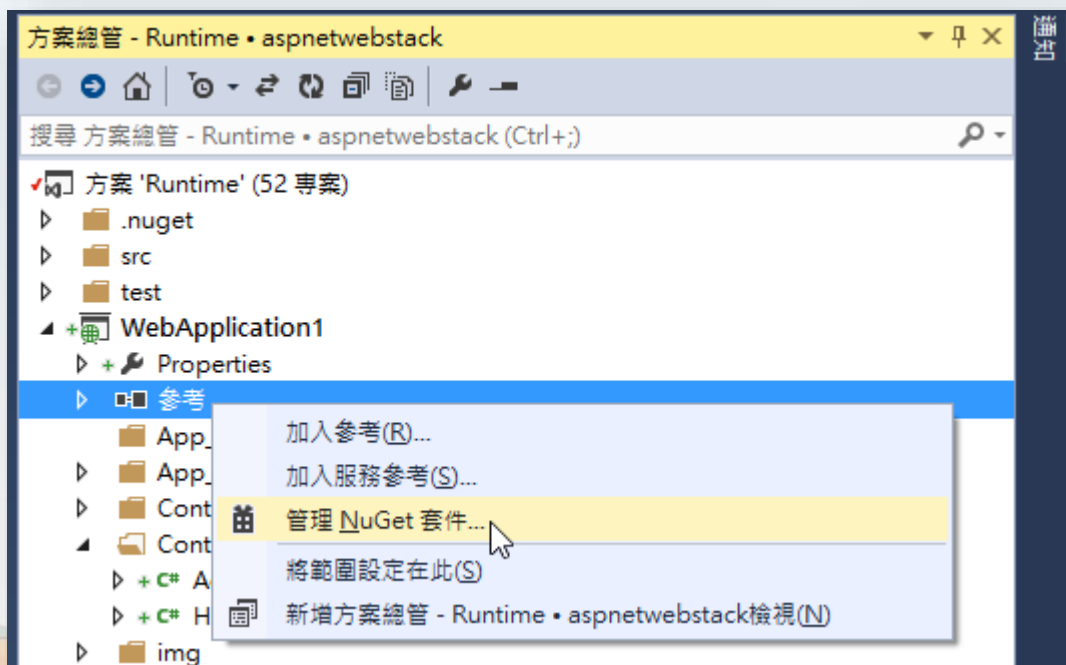
# 管理 NuGet 套件

All services from your imperative.

- ❑ NuGet Package Manager (VS 擴充套件)
- ❑ NuGet Package Explorer (封裝套件的工具)
- ❑ NuGet.exe Command Line (指令列工具)

## ❑ 範例

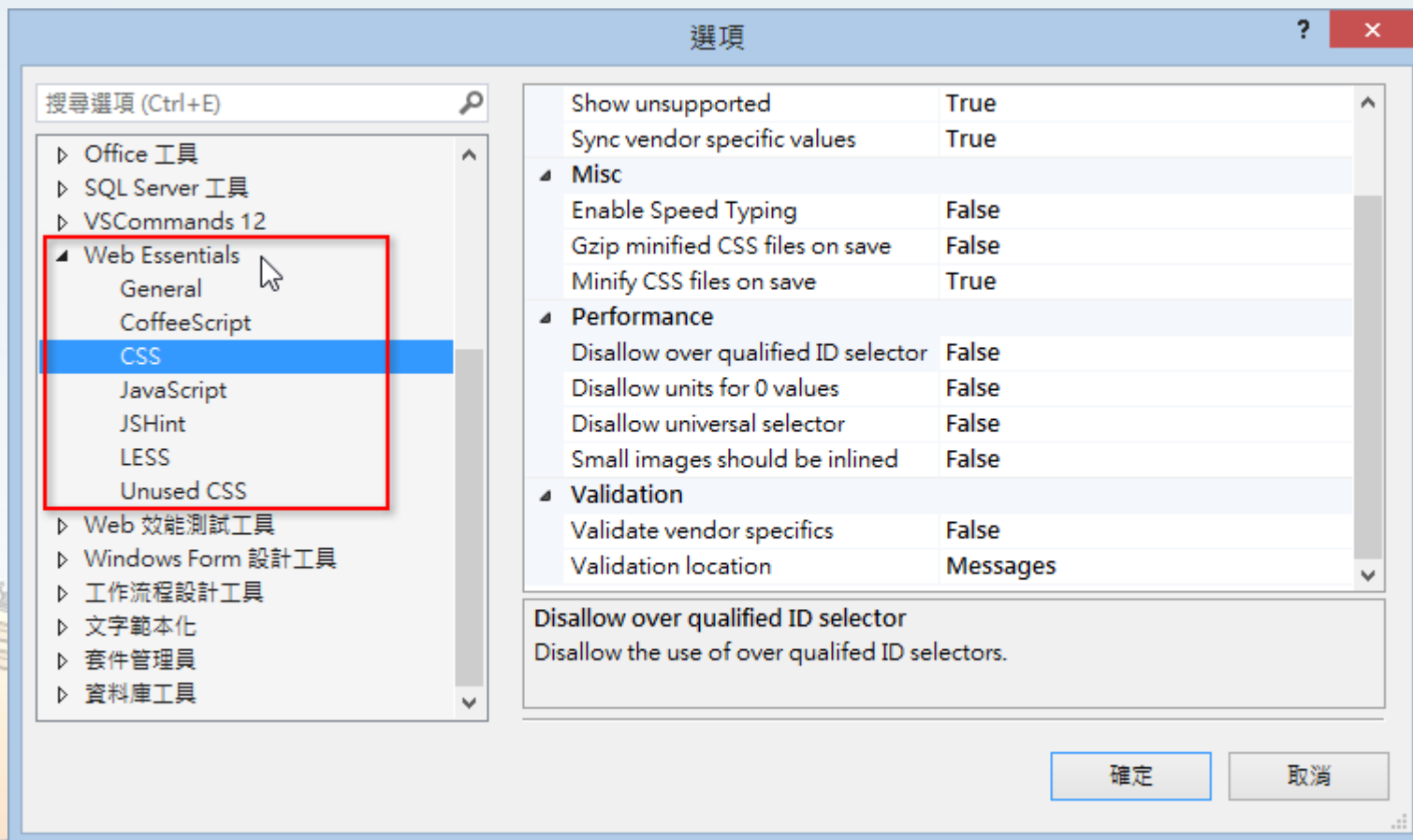
- ASP.NET MVC
- jQuery
- Elmah
- ❑ Elmah.MVC
- ❑ [elmah.io](http://elmah.io)
- NLog



# Web Essentials 2013 (必備套件)

All services from your imperative.

□ <http://vswebessentials.com/>

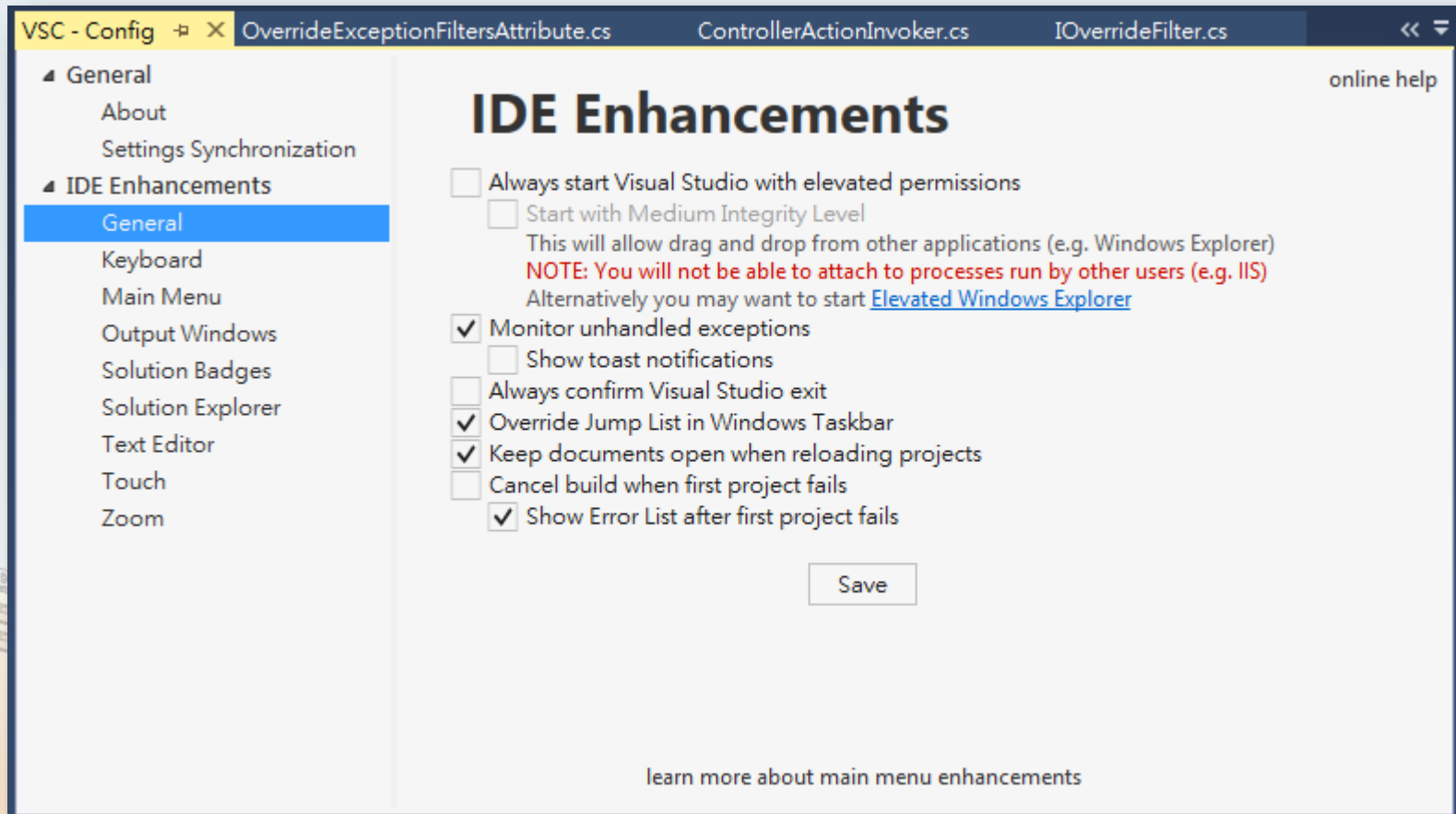




# VSCommands (必備套件)

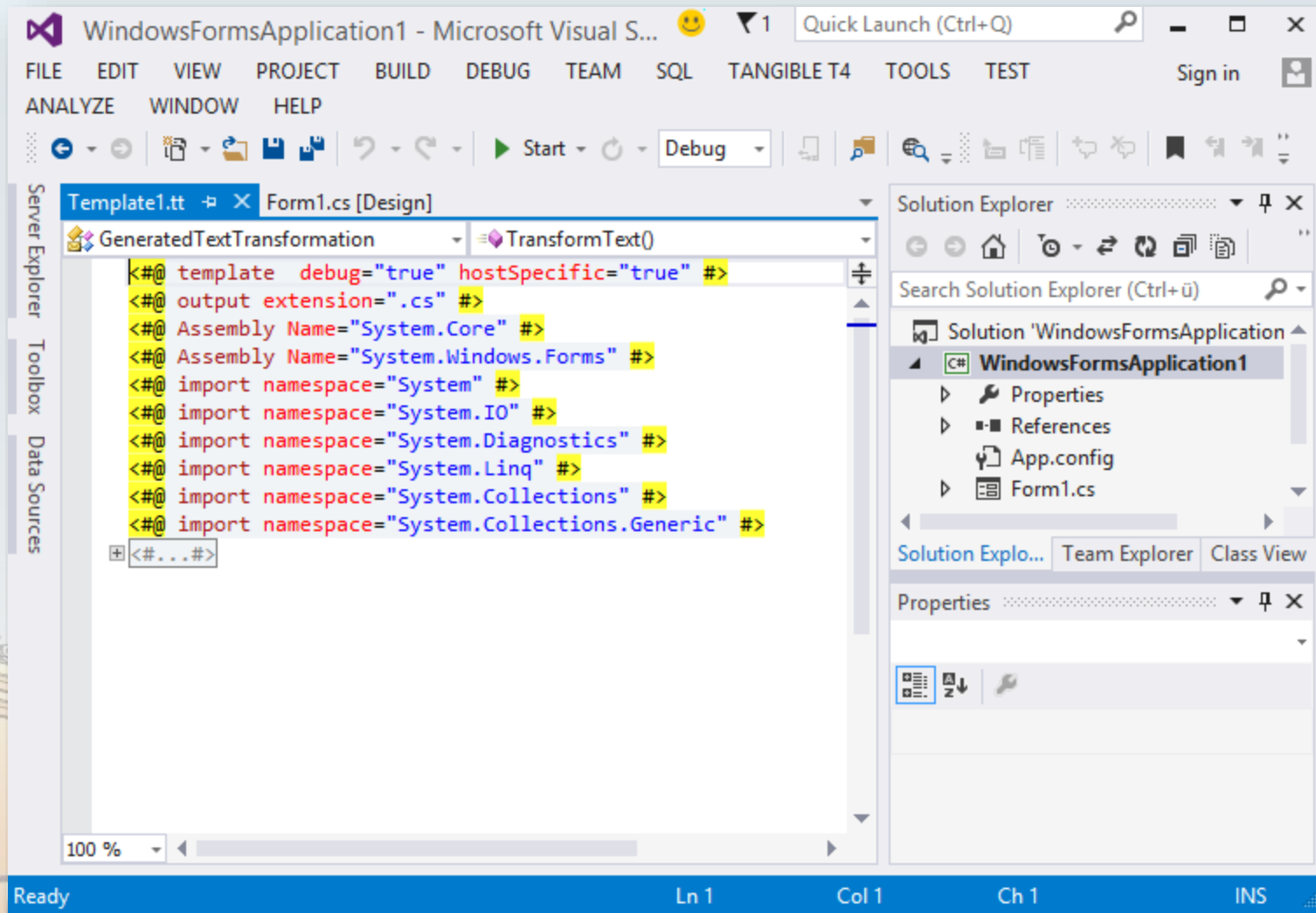
All services from your imperative.

□ <http://vscommands.squaredinfinity.com>



# tangible T4 Editor

All services from your imperative.



# 參考連結

All services from your imperative.

## ☐ Visual Studio 2010 開發技巧文章

- <http://msdn.microsoft.com/zh-tw/vstudio/hh220638.aspx>

## ☐ The Will Will Web | Visual Studio

- <http://blog.miniasp.com/category/Visual-Studio.aspx>

## ☐ The Will Will Web | VS2010 Tips

- <http://blog.miniasp.com/category/VS2010-Tips.aspx>



All services from your imperative.

# ASP.NET MVC 5 開發環境建立



# 安裝必要軟體

All services from your imperative.

- ☐ Visual Studio 2013
- ☐ Visual Studio 2013 Update 3
  
- ☐ Visual Studio 2012
- ☐ Visual Studio 2012 Update 4
  
- ☐ Visual Studio 擴充套件
  - NuGet Package Manager (已內建)
  - Web Essentials 2013
  - VSCommands for Visual Studio 2013
  - tangible T4 Editor

# Visual Studio 2013 範本選擇

All services from your imperative.

## ☐ .NET Framework 4

### ■ 傳統專案範本

- ☐ ASP.NET Web Form 應用程式
- ☐ ASP.NET MVC 4 Web 應用程式

## ☐ .NET Framework 4.5 / 4.5.1

### ■ 剩下一個：ASP.NET Web 應用程式

## ☐ 舊的專案範本

- 整併到 Visual Studio 2012 分類下
- .NET 4.5+：已移除 LINQ to SQL 專案範本！



# 整合 ASP.NET 專案範本

All services from your imperative.

## ☐ 唯一 ASP.NET 專案範本

- ASP.NET Web 應用程式 ( 僅限 .NET Framework 4.5+ )

## ☐ 單一對話框

- 選取範本

- ☐ Empty , Web Forms , MVC , Web API , Single Page Application

- 建立專案可選取身分驗證與授權的方式

<http://www.asp.net/visual-studio/overview/2013/creating-web-projects-in-visual-studio#auth>

- 透過標準方式即可自訂專案範本

<http://www.asp.net/mvc/overview/advanced/custom-mvc-templates>



www.mini-ASP.com





# 介紹 ASP.NET MVC 5 專案範本

All services from your imperative.

## □ 專案範本 (Project Template)

- Empty
- Web Forms
- MVC
- Web API
- Single Page Application

## □ 內建多種常用的套件 ( packages.config )

- 包括 Bootstrap 、 jQuery 、 jQuery.Validation 、 Modernizr 、 Newtonsoft.Json 、 Entity Framework 、 ...
- 套件皆可搭配 NuGet 自動升級!




www.mini-ASP.com


# 介紹 ASP.NET MVC 5 專案範本


All services from your imperative.


新增 ASP.NET 專案 - WebApplication3


選取範本(S):


  
Empty

  
Web Forms

  
MVC

  
Web API

  
Single Page Application

  
Azure Mobile Service

用於建立 ASP.NET MVC 應用程式的專案範本。ASP.NET MVC 可讓您使用「模型-檢視-控制器」架構來建置應用程式。ASP.NET MVC 包括多個功能，可啟用快速且測試驅動的開發，來建立使用最新標準的應用程式。

[深入了解](#)

變更驗證(A)

驗證: 個別使用者帳戶

Microsoft Azure

雲端中的主機(H)

網站

以 wakai@hotmail.com 登入

[管理訂閱](#)

為下列項目新增資料夾及核心參考:

☐ Web Forms ☒ MVC ☐ Web API

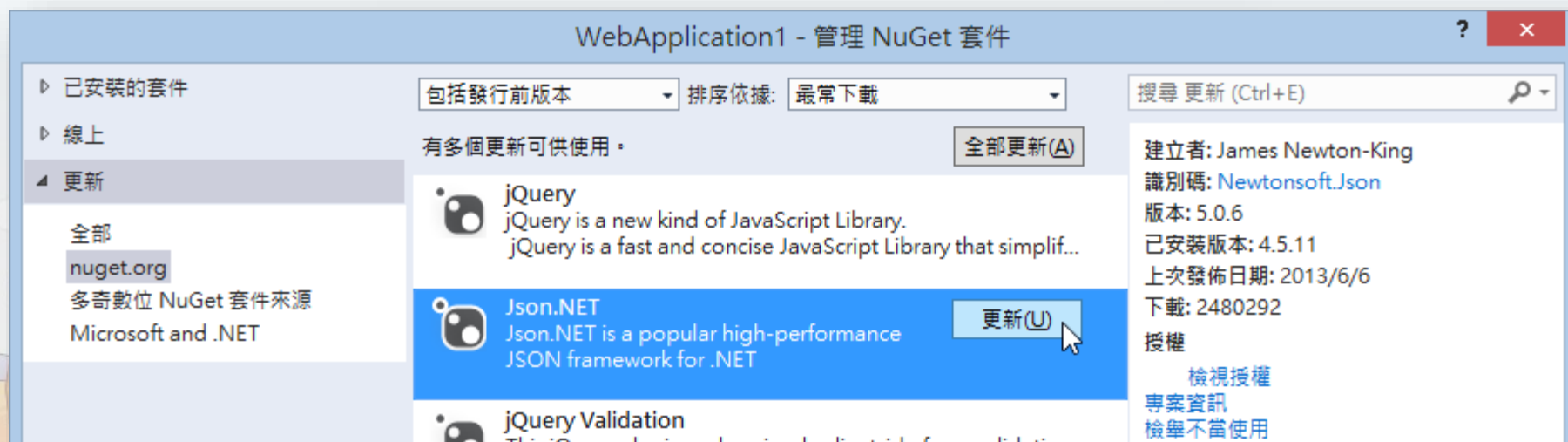
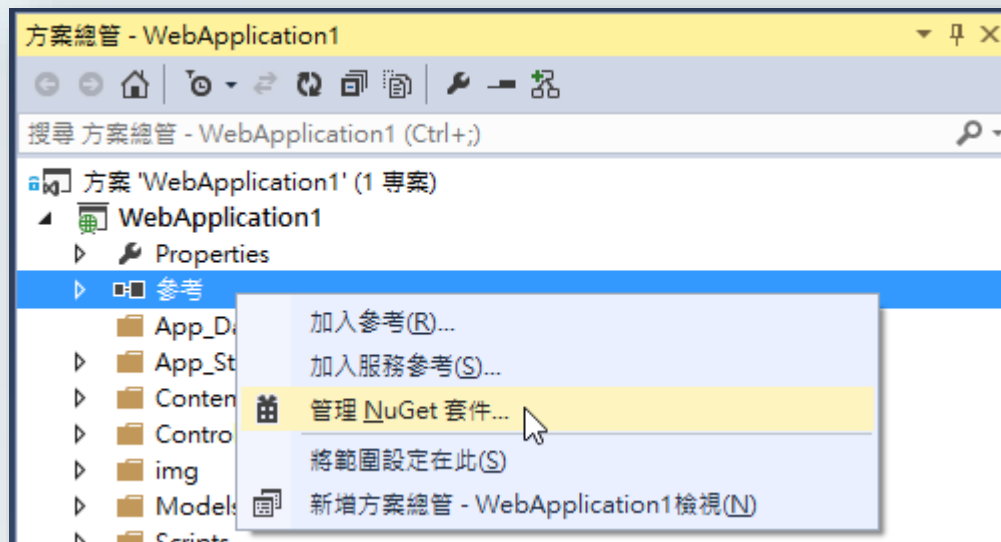
☐ 新增單元測試

測試專案名稱: WebApplication3.Tests

確定 取消

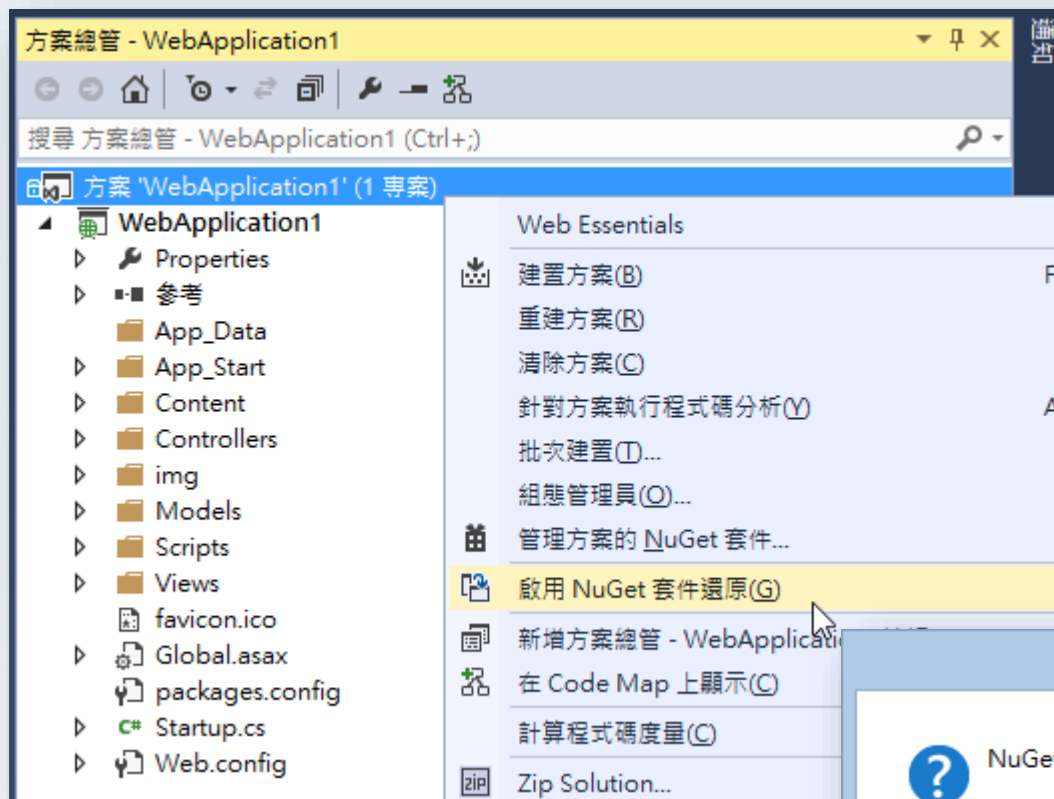
# 透過 NuGet 升級內建套件

All services from your imperative.



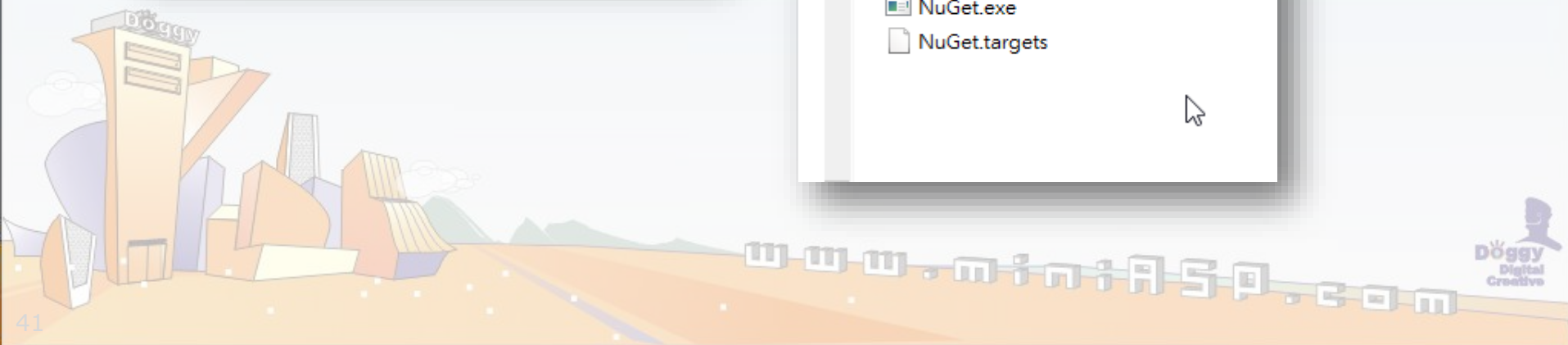
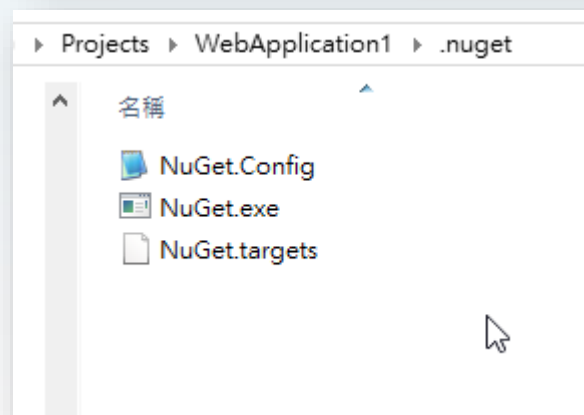
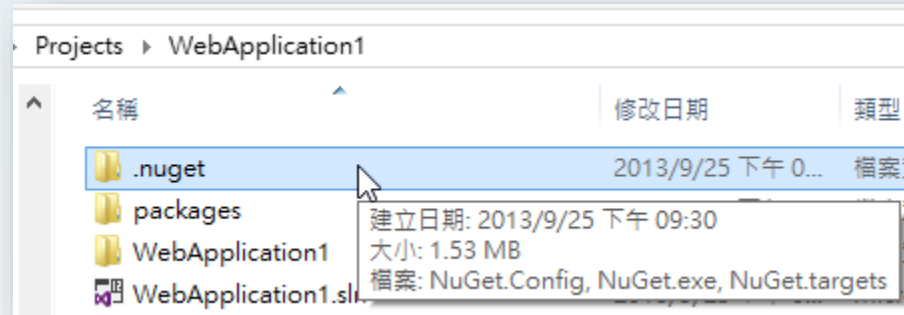
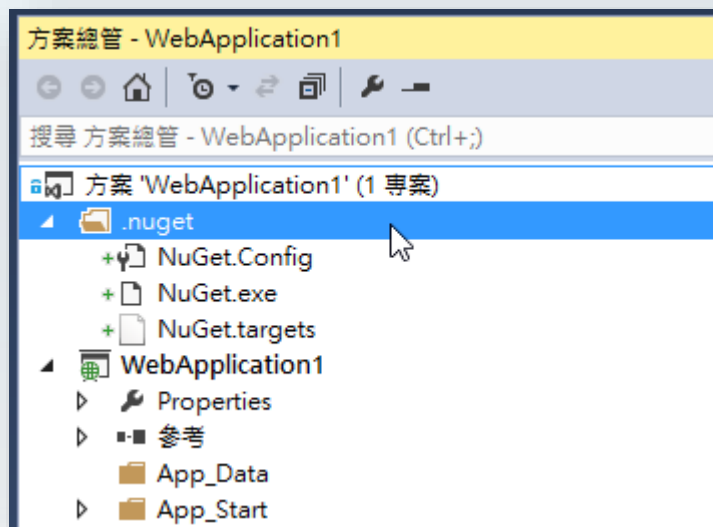
# 啟用 NuGet 套件還原

All services from your imperative.



# 啟用 NuGet 套件還原

All services from your imperative.



# 專案範本結構介紹

All services from your imperative.

## □ 主要目錄

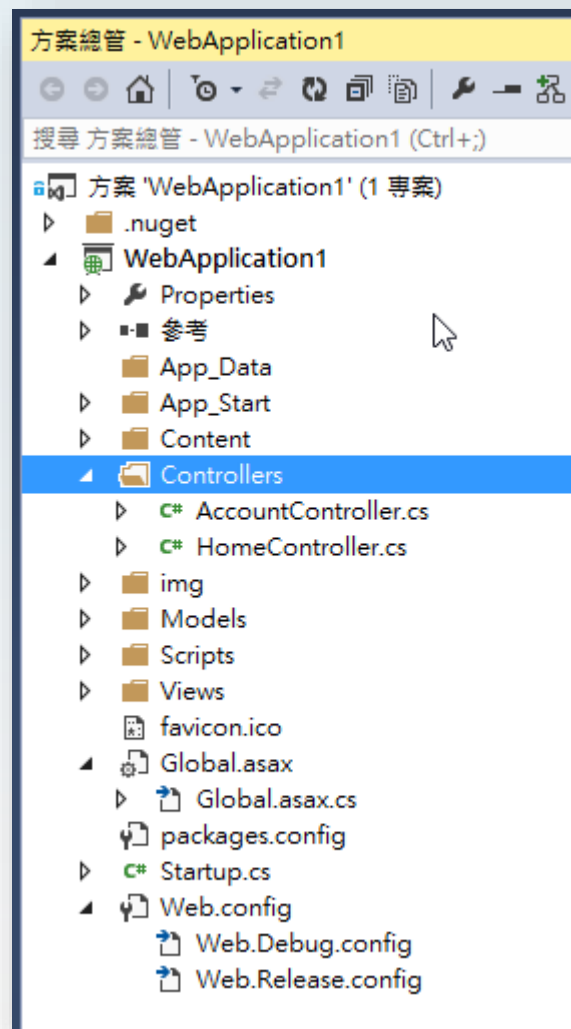
- Controllers
- Models
- Views
- **App\_Start**

## □ 資源檔案目錄

- App\_Data
- Content
- Scripts

## □ 重要檔案

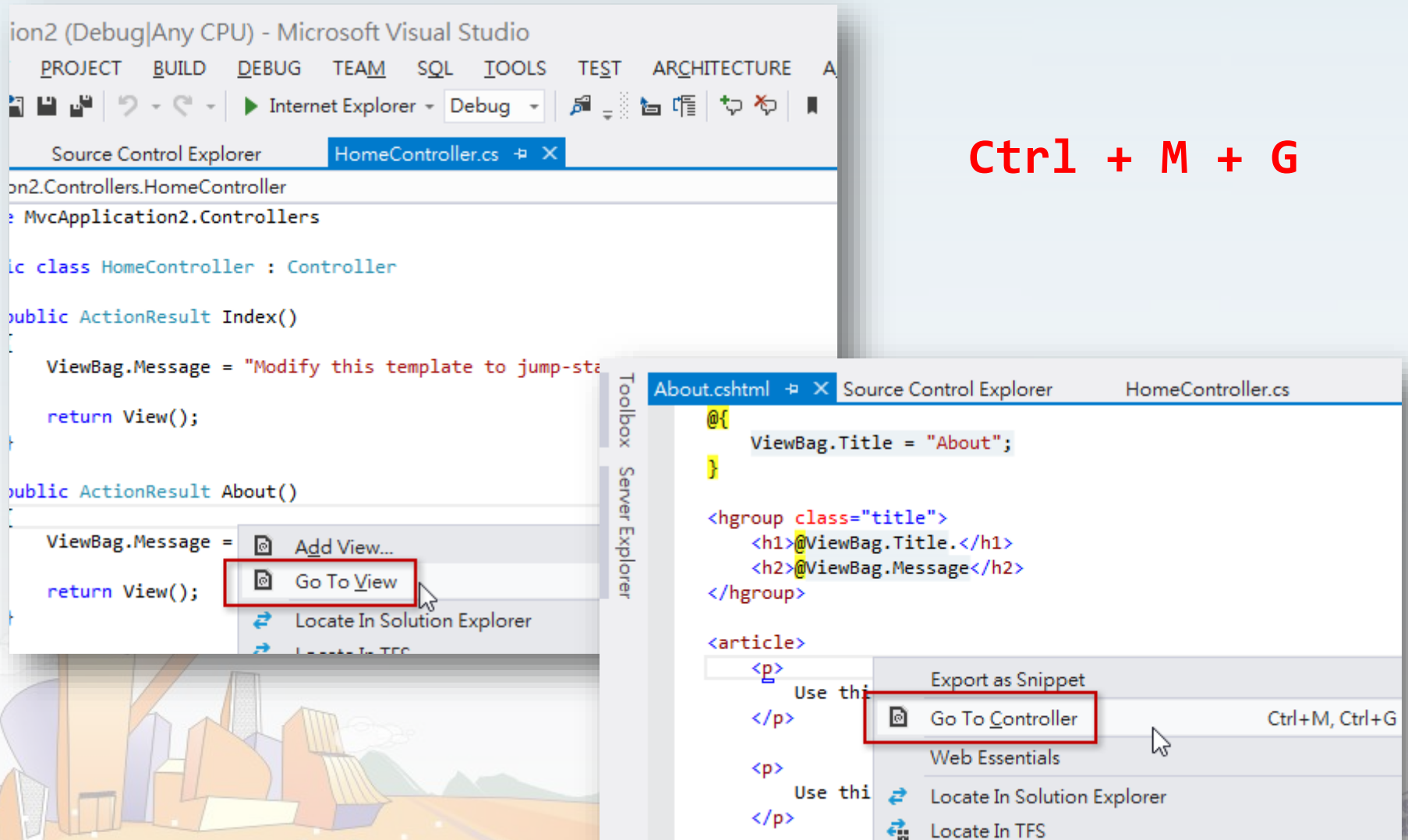
- Web.config
- Global.asax.cs
- packages.config
- Startup.cs



# 快速於 Controller 與 View 之間切換

All services from your imperative.

**Ctrl + M + G**







# ASP.NET MVC 5 開發實戰

## ASP.NET MVC 架構與觀念解析



多奇數位創意有限公司

技術總監 黃保翕 (Will 保哥)

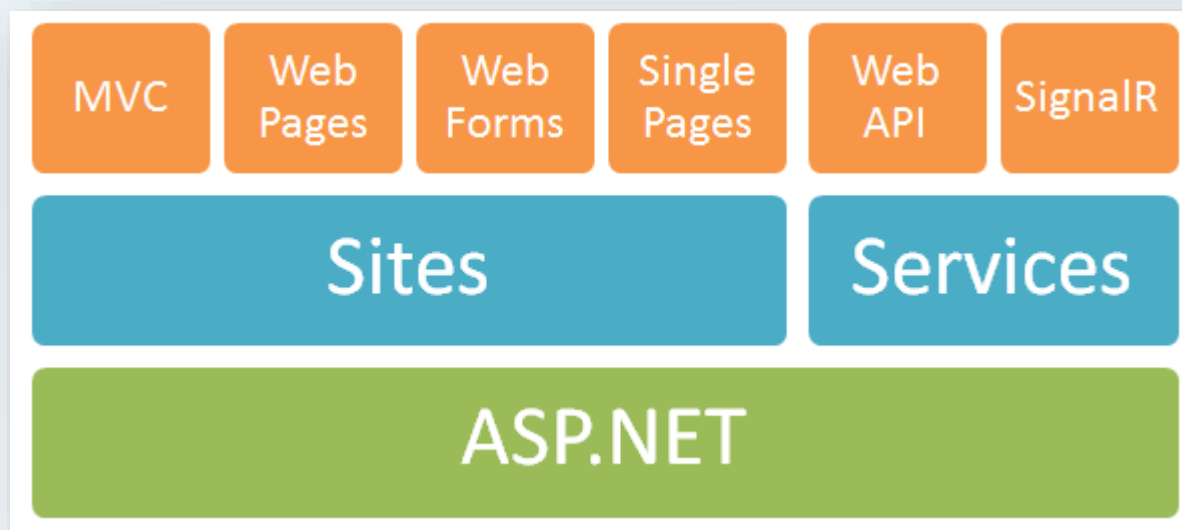
部落格：<http://blog.miniasp.com/>



# 瞭解 ASP.NET 架構與發展狀況

All services from your imperative.

## □ ASP.NET 發展現況



All services from your imperative.

# WEB FORM 與 MVC 的差異



# Web Form 與 ASP.NET MVC 的差異

All services from your imperative.

支援功能	ASP.NET Web Form	ASP.NET MVC
ViewState	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ASP.NET 頁面追蹤機制 (Page Trace)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ASP.NET 事件驅動模型 (Event Model)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
伺服器控制項 (Server Control)	<input checked="" type="checkbox"/>	部分支援
System.Web.SiteMapProvider 類別	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>ASP.NET Provider Model</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>System.Web.Caching</b> 命名空間	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>System.Web.SessionState</b> 命名空間	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>System.Web.Security</b> 命名空間	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>Profile, Membership, SiteMap</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
其他 <b>System.Web.*</b> 功能	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

# Web Form 與 ASP.NET MVC 的差異

All services from your imperative.

## □ 相同之處

- 都是 IHttpHandler 的產物
- 共享 ASP.NET Framework 的所有功能

## □ IL 指令集差很多

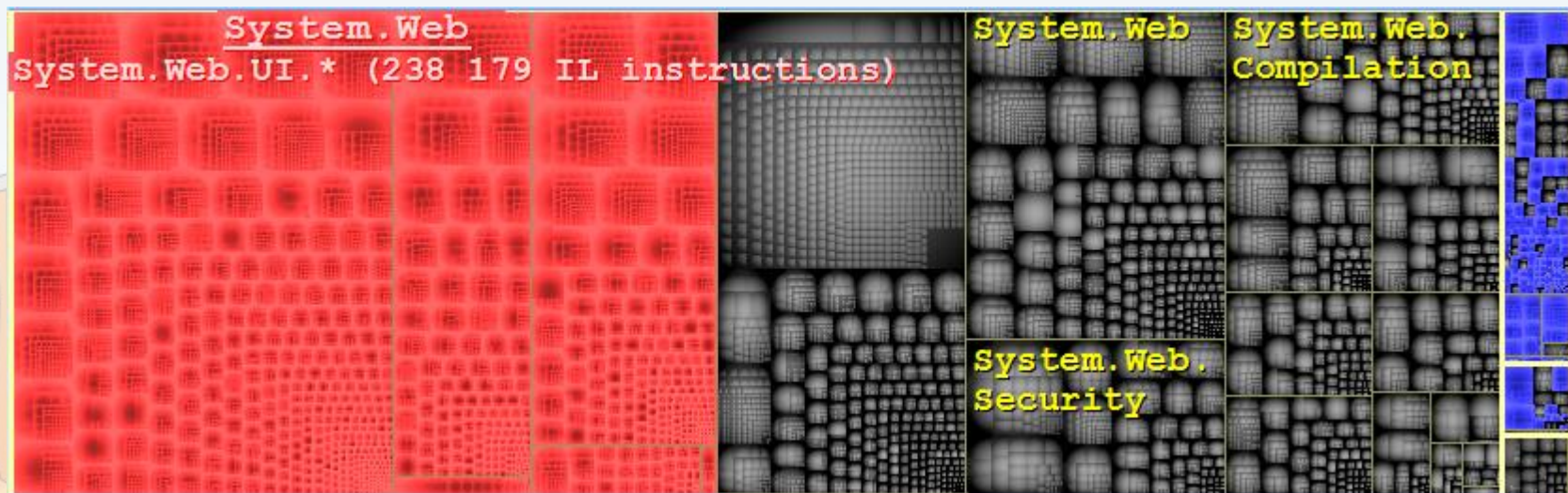
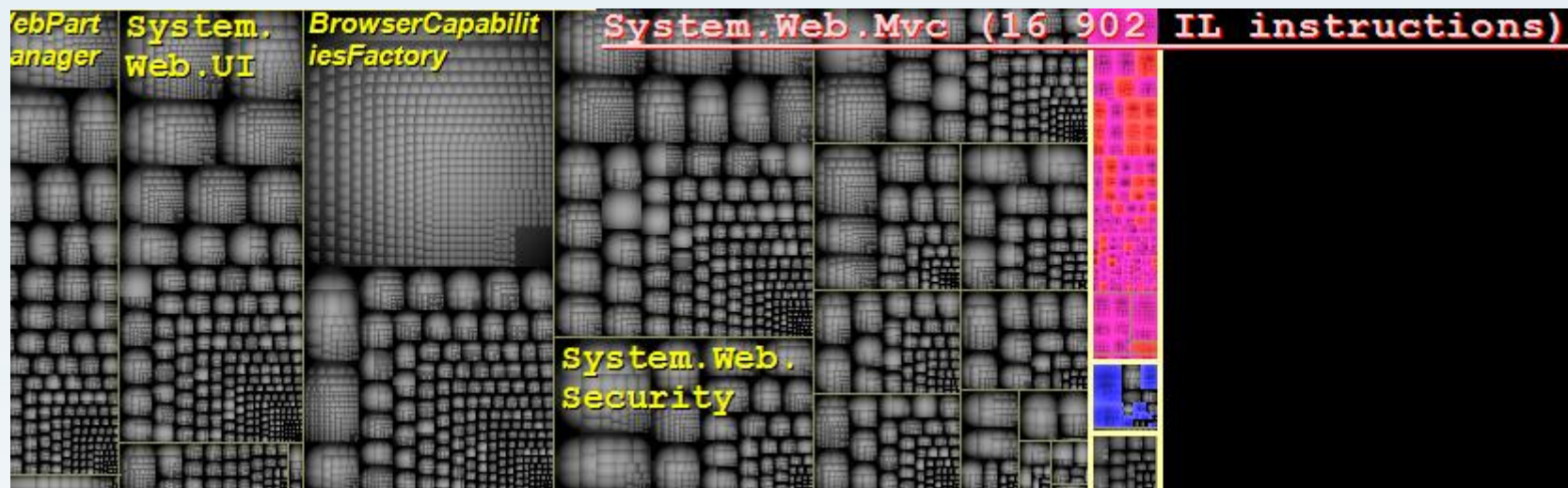
- System.Web.Mvc.\*: 16,902 IL instructions
  - 雖然 ASP.NET MVC 的 WebFormView 用到許多原本 WebForm 的東西，但是還是很多控制項不會去用他！
  - ASP.NET MVC 的 ViewPage 繼承 System.Web.UI.Page (非常邪惡)
- System.Web.UI.\*: 238,179 IL instructions





# 指令集數量的差別

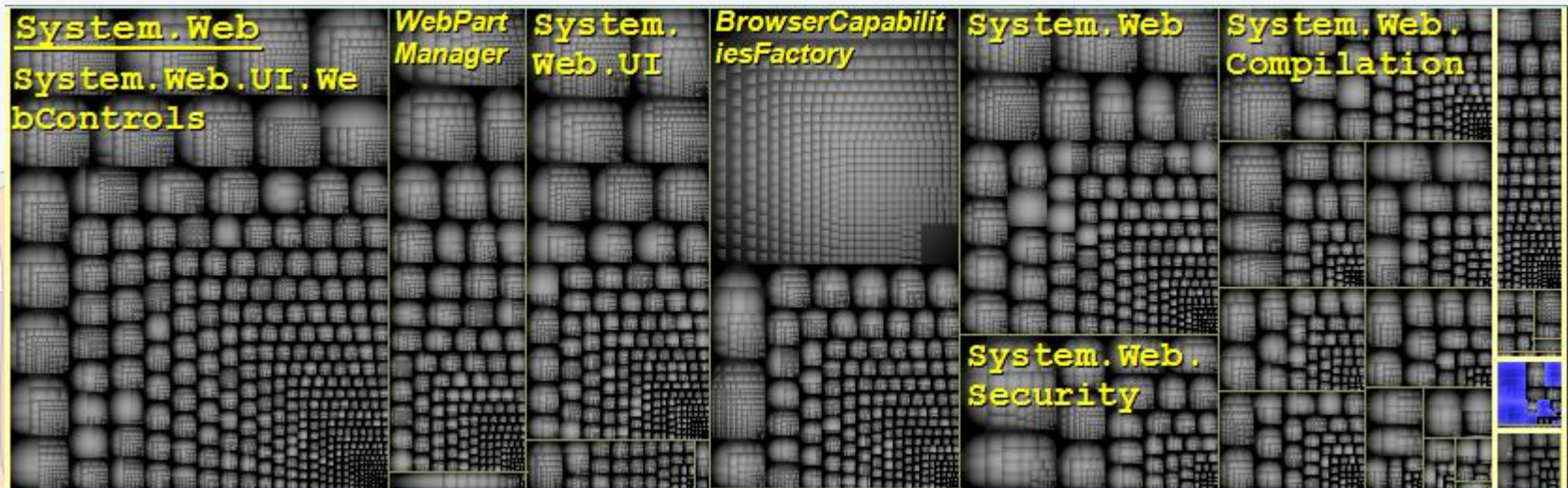
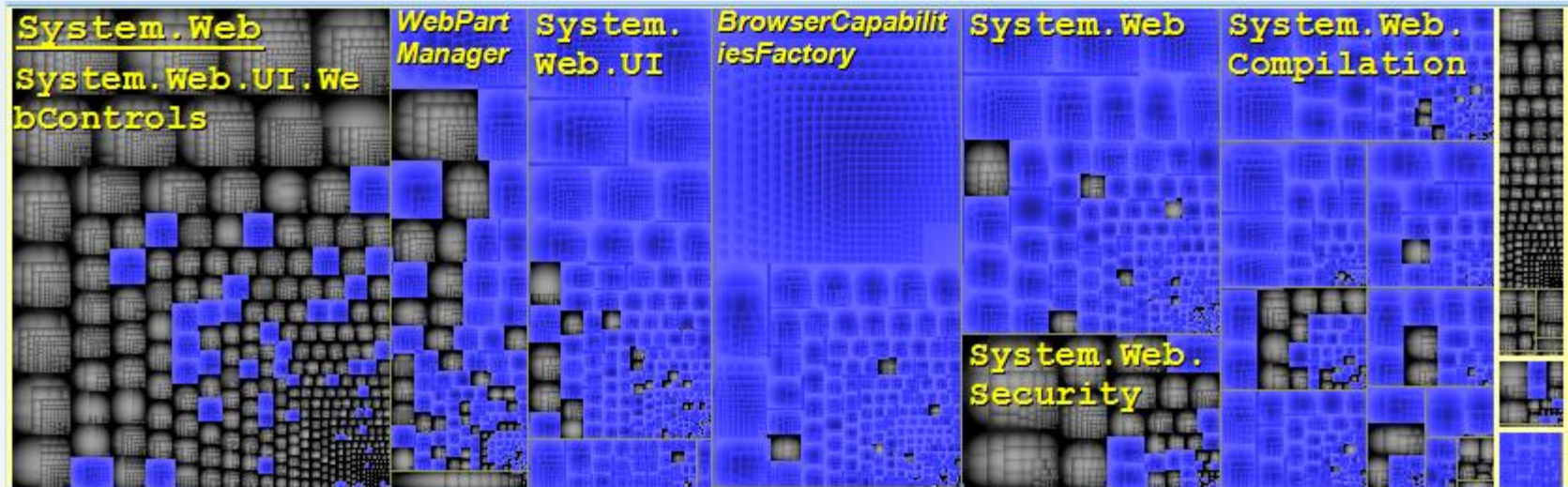
All services from your imperative.





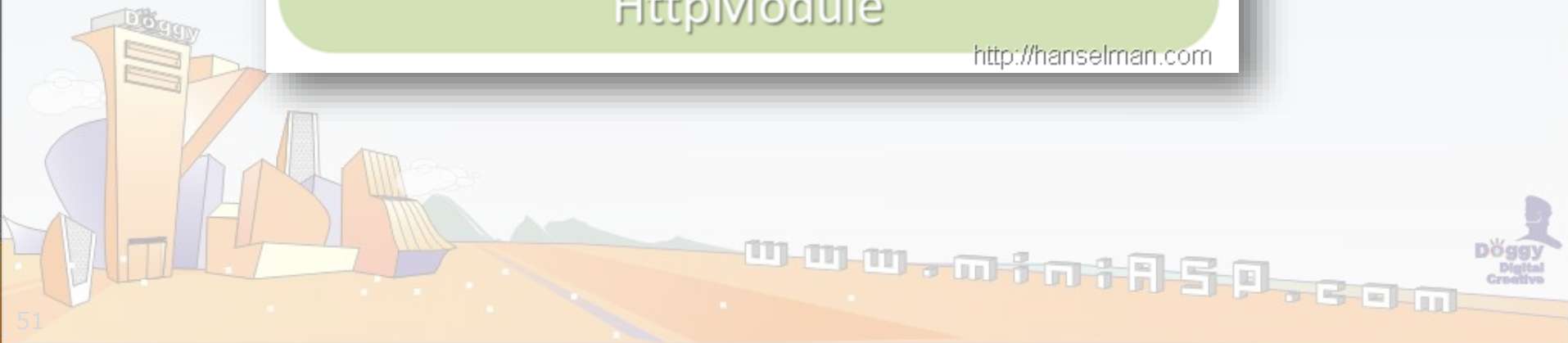
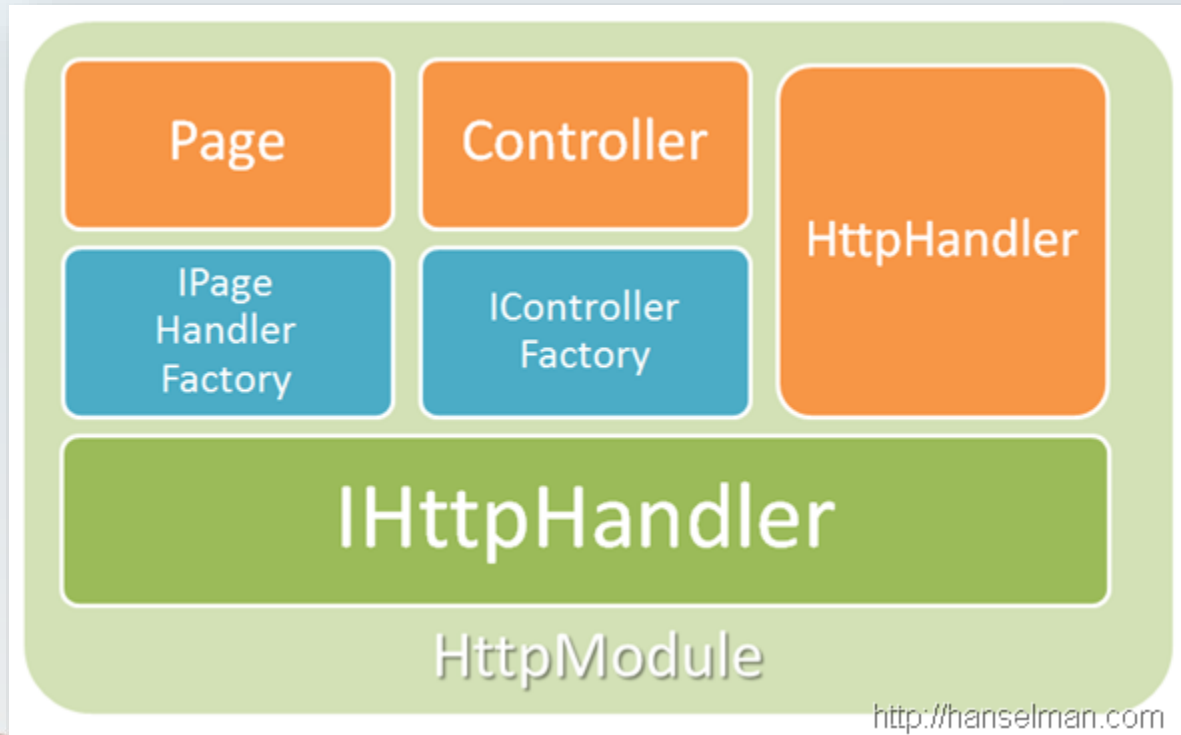
# 有沒有 WebFormView 的差別

All services from your imperative.



# HttpModule 與 HttpHandler

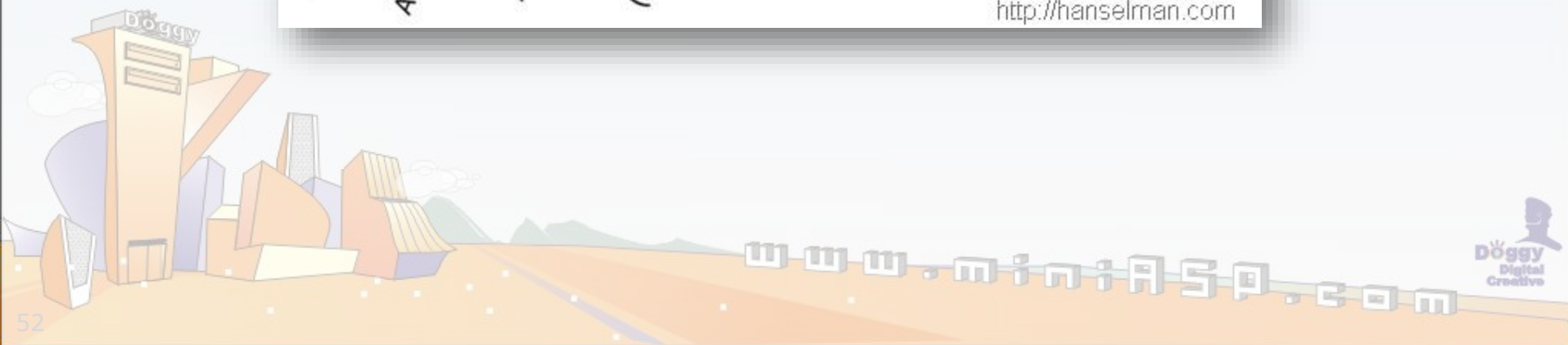
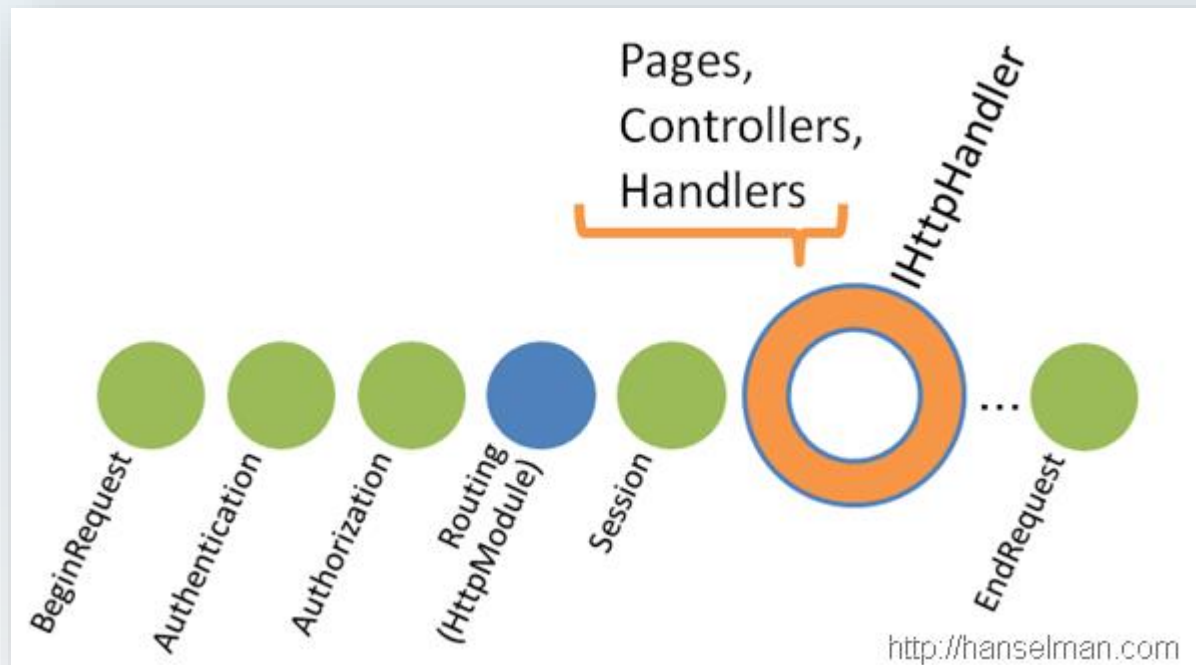
All services from your imperative.






# ASP.NET 執行的過程

All services from your imperative.



# WebForm 的執行過程

All services from your imperative.

Call Stack	
Name	
 App_Web_4mzlopy8.dll!WebForm.Page_Load(object sender = {ASP.webform_aspx}, System.EventArgs e = {System.EventArgs}) Line 12	
System.Web.RegularExpressions.dll!System.Web.Util.CalliHelper.EventArgFunctionCaller(System.IntPtr fp, object o, object t, System.EventArgs e)	
System.Web.dll!System.Web.Util.CalliEventHandlerDelegateProxy.Callback(object sender, System.EventArgs e) + 0x24 bytes	
System.Web.dll!System.Web.UI.Control.OnLoad(System.EventArgs e) + 0x64 bytes	
System.Web.dll!System.Web.UI.Control.LoadRecursive() + 0x33 bytes	
System.Web.dll!System.Web.UI.Page.ProcessRequestMain(bool includeStagesBeforeAsyncPoint = true, bool includeStagesAfterAsyncPoint = true)	
System.Web.dll!System.Web.UI.Page.ProcessRequest(bool includeStagesBeforeAsyncPoint, bool includeStagesAfterAsyncPoint = true) + 0x84 bytes	
System.Web.dll!System.Web.UI.Page.ProcessRequest() + 0x51 bytes	
System.Web.dll!System.Web.UI.Page.ProcessRequestWithNoAssert(System.Web.HttpContext context) + 0x16 bytes	
System.Web.dll!System.Web.UI.Page.ProcessRequest(System.Web.HttpContext context) + 0x32 bytes	
App_Web_4mzlopy8.dll!ASP.webform_aspx.ProcessRequest(System.Web.HttpContext context = {System.Web.HttpContext}) + 0x33 bytes	
System.Web.dll!System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute() + 0xb6 bytes	
System.Web.dll!System.Web.HttpApplication.ExecuteStep(System.Web.HttpApplication.IExecutionStep step = {System.Web.HttpApplication.CallHandlerExecutionStep}) + 0x10 bytes	
System.Web.dll!System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(System.Exception error) + 0x133 bytes	
System.Web.dll!System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(System.Web.HttpContext context, System.AsyncCallback callback, object state) + 0x10 bytes	
System.Web.dll!System.Web.HttpRuntime.ProcessRequestInternal(System.Web.HttpWorkerRequest wr = {Microsoft.VisualStudio.WebHost.Request}) + 0x10 bytes	
System.Web.dll!System.Web.HttpRuntime.ProcessRequestNoDemand(System.Web.HttpWorkerRequest wr) + 0x63 bytes	
System.Web.dll!System.Web.HttpRuntime.ProcessRequest(System.Web.HttpWorkerRequest wr) + 0x47 bytes	
WebDev.WebHost.dll!Microsoft.VisualStudio.WebHost.Request.Process() + 0x165 bytes	
WebDev.WebHost.dll!Microsoft.VisualStudio.WebHost.Host.ProcessRequest(Microsoft.VisualStudio.WebHost.Connection conn = {System.Runtime.Remoting.Channels.Local.LocalConnection}) + 0x10 bytes	
[Appdomain Transition]	
WebDev.WebHost.dll!Microsoft.VisualStudio.WebHost.Server.OnSocketAccept(object acceptedSocket) + 0x7d bytes	
mscorlib.dll!System.Threading._ThreadPoolWaitCallback.WaitCallback_Context(object state) + 0x2f bytes	
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object state) + 0x17 bytes	
mscorlib.dll!System.Threading._ThreadPoolWaitCallback.PerformWaitCallbackInternal(System.Threading._ThreadPoolWaitCallback tpWaitCallback) + 0x10 bytes	
mscorlib.dll!System.Threading._ThreadPoolWaitCallback.PerformWaitCallback(object state) + 0x59 bytes	

# ASP.NET MVC 的執行過程

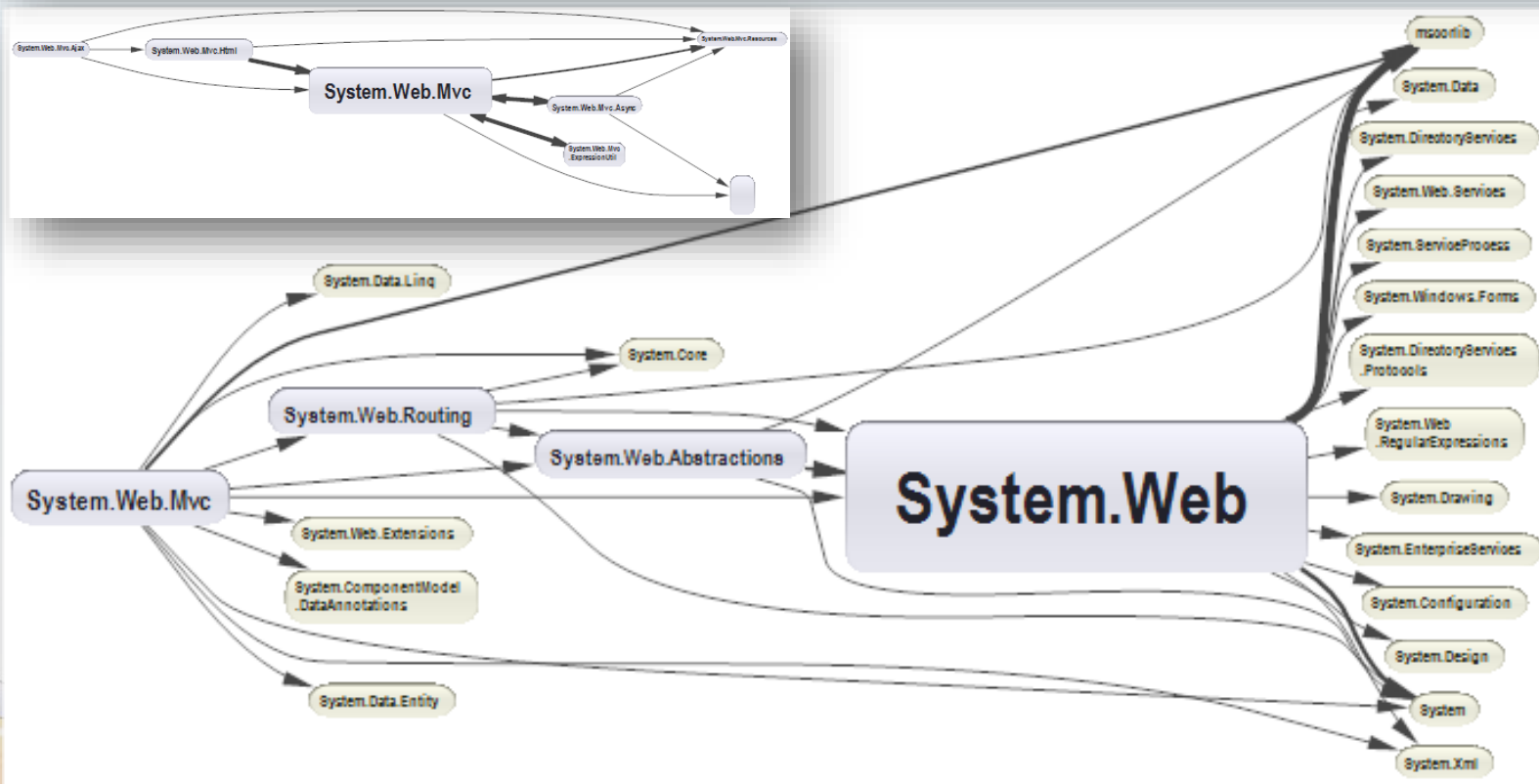
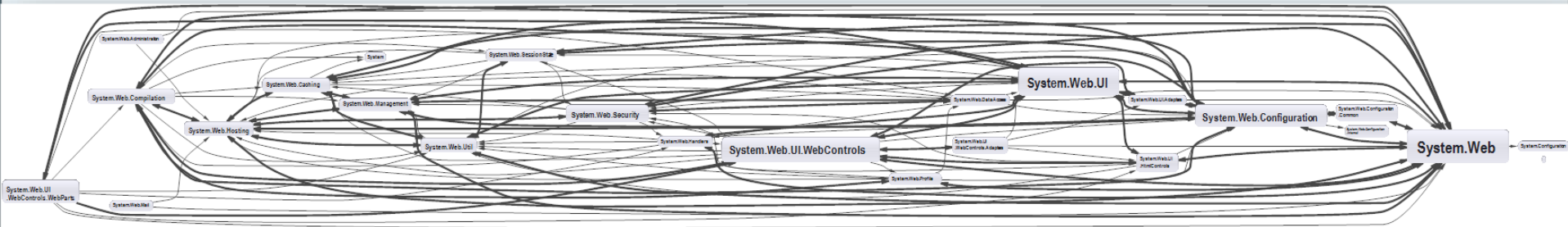
All services from your imperative.

Call Stack	
Name	
App_Code.16bw--2k.dll!MvcApplication1.Controllers.HomeController.About() Line 21	
[Lightweight Function]	
System.Web.Mvc.dll!System.Web.Mvc.ActionMethodDispatcher.Execute(System.Web.Mvc.ControllerBase controller, object[] parameters) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ReflectedActionDescriptor.Execute(System.Web.Mvc.ControllerContext controllerContext, System.Collections.Generic.IDictionary`2 parameters) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerActionInvoker.InvokeActionMethod(System.Web.Mvc.ControllerContext controllerContext = {System.Web.Mvc.ControllerContext}) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerActionInvoker.InvokeActionMethodWithFilters.AnonymousMethod() + 0x35 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerActionInvoker.InvokeActionMethodFilter(System.Web.Mvc.IActionFilter filter = {MvcApplication1.Controllers.HomeController.About()}) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerActionInvoker.InvokeActionMethodWithFilters.AnonymousMethod() + 0x14 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerActionInvoker.InvokeActionMethodWithFilters(System.Web.Mvc.ControllerContext controllerContext = {System.Web.Mvc.ControllerContext}) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerActionInvoker.InvokeAction(System.Web.Mvc.ControllerContext controllerContext = {System.Web.Mvc.ControllerContext}) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.Controller.ExecuteCore() + 0x7f bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerBase.Execute(System.Web.Routing.RequestContext requestContext) + 0x1c bytes	
System.Web.Mvc.dll!System.Web.Mvc.ControllerBase.System.Web.Mvc.IController.Execute(System.Web.Routing.RequestContext requestContext) + 0x12 bytes	
System.Web.Mvc.dll!System.Web.Mvc.MvcHandler.ProcessRequest(System.Web.HttpContextBase httpContext) + 0x98 bytes	
System.Web.Mvc.dll!System.Web.Mvc.MvcHandler.ProcessRequest(System.Web.HttpContext httpContext) + 0x3a bytes	
System.Web.Mvc.dll!System.Web.Mvc.MvcHandler.System.Web.IHttpHandler.ProcessRequest(System.Web.HttpContext httpContext) + 0x8 bytes	
System.Web.dll!System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute() + 0xb6 bytes	
System.Web.dll!System.Web.HttpApplication.ExecuteStep(System.Web.HttpApplication.IExecutionStep step = {System.Web.HttpApplication.CallHandlerExecutionStep}) + 0x12 bytes	
System.Web.dll!System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(System.Exception error) + 0x133 bytes	
System.Web.dll!System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(System.Web.HttpContext context, System.AsyncCallback callback, object state) + 0x12 bytes	
System.Web.dll!System.Web.HttpRuntime.ProcessRequestInternal(System.Web.HttpWorkerRequest wr = {Microsoft.VisualStudio.WebHost.Request}) + 0x12 bytes	
System.Web.dll!System.Web.HttpRuntime.ProcessRequestNoDemand(System.Web.HttpWorkerRequest wr) + 0x63 bytes	
System.Web.dll!System.Web.HttpRuntime.ProcessRequest(System.Web.HttpWorkerRequest wr) + 0x47 bytes	
WebDev.WebHost.dll!Microsoft.VisualStudio.WebHost.Request.Process() + 0x165 bytes	
WebDev.WebHost.dll!Microsoft.VisualStudio.WebHost.Host.ProcessRequest(Microsoft.VisualStudio.WebHost.Connection conn = {System.Runtime.Remoting.Channels.Local.LocalConnection}) + 0x12 bytes	
[Appdomain Transition]	
WebDev.WebHost.dll!Microsoft.VisualStudio.WebHost.Server.OnSocketAccept(object acceptedSocket) + 0x7d bytes	
mscorlib.dll!System.Threading._ThreadPoolWaitCallback.WaitCallback_Context(object state) + 0x2f bytes	
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object state) + 0x12 bytes	
mscorlib.dll!System.Threading._ThreadPoolWaitCallback.PerformWaitCallbackInternal(System.Threading._ThreadPoolWaitCallback tpWaitCallback) + 0x12 bytes	
mscorlib.dll!System.Threading.ThreadPoolWaitCallback.PerformWaitCallback(object state) + 0x59 bytes	



# Web Form 與 ASP.NET MVC 的差異

# All services from your imperative.



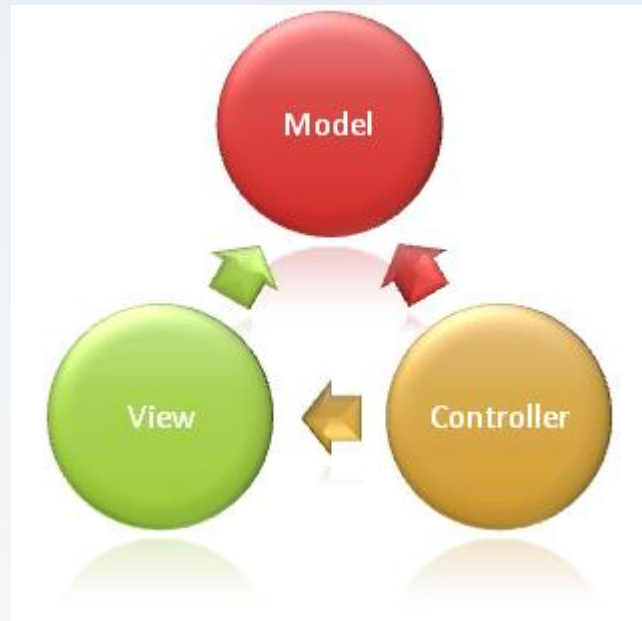
All services from your imperative.

# ASP.NET MVC 核心觀念



# 瞭解 MVC 設計樣式

All services from your imperative.



# 關注點分離 (Separate of Concerns)

All services from your imperative.

## □ 簡化複雜度

- 將複雜的問題拆解成數個容易解決的單元
- 一次僅關注於一個較容易理解與解決的部分

## □ 可維護性大幅提昇

- 不止可區分 Model、View、Controller
- 還可以再切割成更多層次，讓專案更易於維護

## □ 更容易測試

- 僅測試需要關注的點
- 測試導向先行 (Test-Driven Development; TDD)





# 以習慣取代配置

All services from your imperative.

## □ 約定俗成的開發習慣

- 能將繁瑣的開發流程簡化
- 有效減少溝通與決策的時間，降低溝通成本

## □ 培養團隊的習慣

- 瞭解 ASP.NET MVC 的開發習慣極其重要
- 開發團隊能有自己的開發習慣
  - 方案規劃、命名空間、命名規則、目錄結構
  - 使用 NuGet、Entity Framework、EFRepository、MVC Turbine、...
  - 不要在 View 裡使用 LINQ to SQL 的「點表示法」( dot notation )
  - 適度的使用 ViewModel





# 控制反轉 (Inversion of Control)

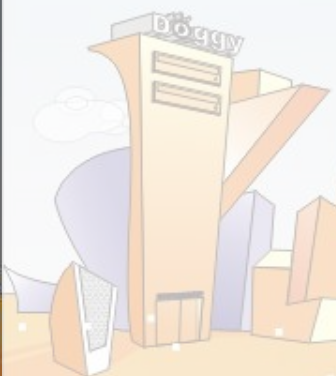
All services from your imperative.

## □ 專注於軟體設計

- 讓各模組之間透過 **合約** (contracts) 進行合作
- 各模組無法對其他模組進行**假設**，因而專注於實作
- 組合模組的人可專注於**介面設計**，而非**實作細節**
- 替換任何模組時不會有**副作用** (side effect)

## □ 特性

- 執行時期 (runtime) 才會注入 (injects) 實體
- 透過**相依性注入** (Dependency injection) 實現 IoC
- 較容易落實「關注點分離」但也容易變複雜



# ASP.NET MVC 擴充點 - Model

All services from your imperative.

## □ 擴充點 (extensibility points)

### ■ Model

- ValueProvider           輸入數值轉換成資料
- ModelBinder   輸入資料轉換成型別
  - DefaultModelBinder 不支援 immutable objects
- ModelMetadata
- DataAnnotation / IValidatableObject
  - ValidationAttribute
    - DataAnnotationsModelValidatorProvider
  - IDataErrorInfo
    - DataErrorInfoModelValidatorProvider
  - ClientDataTypeModelValidatorProvider
  - ModelValidatorProvider



# ASP.NET MVC 擴充點 - View

All services from your imperative.

## ☐ 擴充點 (extensibility points)

### ■ View

#### ☐ ViewEngine : VirtualPathProviderViewEngine

##### ■ WebFormViewEngine

##### ■ RazorViewEngine

#### ☐ HTML Helper

#### ☐ Razor Helper

### ■ Controller



# ASP.NET MVC 擴充點 - Controller

All services from your imperative.

## □ 擴充點 (extensibility points)

### ■ Controller

#### □ Action Selectors

- [ActionName]
- [ActionMethodSelector]

#### □ Action Filters

- Authorization Filters
  - [Authorize]
- Exception Filters
  - [HandleError]
- Result Filter
  - [OutputCache]

#### □ ControllerFactory

#### □ ControllerBase

#### □ Action Results

### ■ Routing

#### □ RouteConstraint

#### □ RouteHandler



# 分層開發

All services from your imperative.

## □ 透過 MVC 分階段開發

- Model → Controller → View
- Model → Repository → Controller → View
- Model → Repository → Service → Controller → View
- IoC / DI / AutoMapper / Unit Test / Integration Test

## □ 透過關注點分離簡化複雜度

- 漸進式擴充各 ASP.NET MVC 擴充點
- HTML Helper, ActionFilter, Model Binder, Model Validation, ...

## □ 定時做 Code Refactoring

- 推薦書籍：重構 - 向範式前進 (Refactoring to Patterns)



# 團體合作的ASP.NET MVC開發策略

All services from your imperative.

## □ 架構師專注 Model

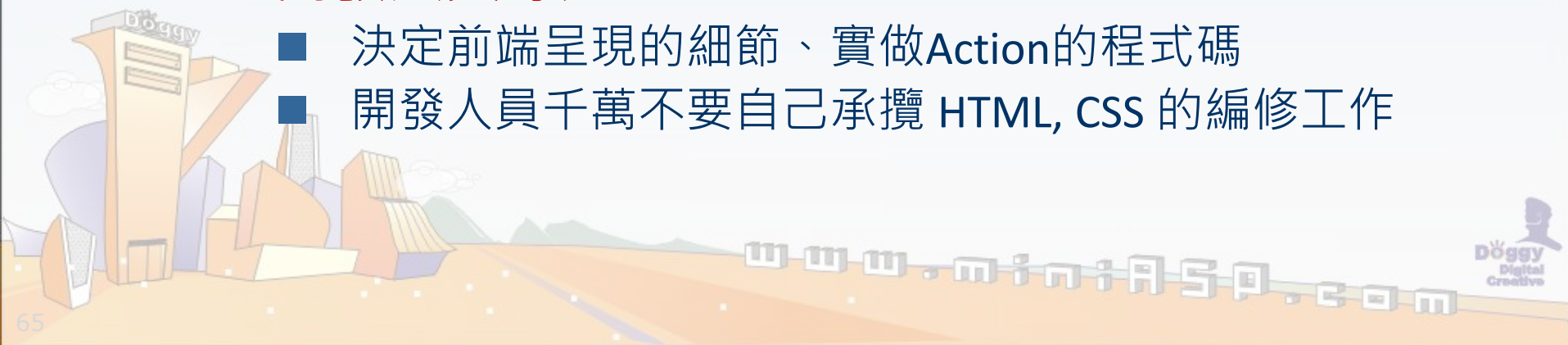
- 設計資料結構、資料驗證規則、商業邏輯
- LINQ to SQL, Repository, Service, DAL
- Model Validation, Model Binder, DataAnnotation

## □ 分析師專注 Controller

- 設計網頁與後端的互動關係
- URL Routing, Controller, Actions, Action Filter
- 撰寫單元測試程式 (非整合測試)

## □ 開發人員專注 View

- 決定前端呈現的細節、實做Action的程式碼
- 開發人員千萬不要自己承攬 HTML, CSS 的編修工作



All services from your imperative.

# ASP.NET MVC 執行生命週期



# ASP.NET MVC 執行生命週期(簡化版)

All services from your imperative.

HTTP 要求

- `http://example.com/Product/Detail/1`
- `GET /Product/Detail/1`

Controller

- `ProductController.Detail(1)`

Model

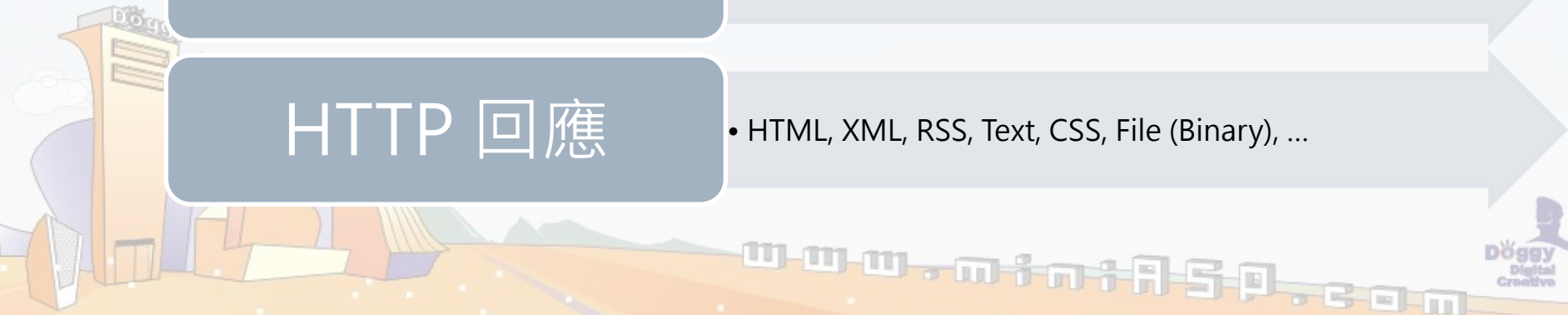
- `ProductRepository.GetProduct(1)`

View

- `~/Views/Product/Detail.aspx`

HTTP 回應

- HTML, XML, RSS, Text, CSS, File (Binary), ...





# ASP.NET MVC 執行生命週期(簡化版)

All services from your imperative.

HTTP 要求

- POST /Product/Edit/1

Controller

- ProductController.Edit(1, FormCollection c)
- TryUpdateModel(product, new string[] { "Name"})

Model

- var product = ProductRepository.GetProduct(1)
- ProductRepository.UpdateProduct(1, product)

View

- N/A

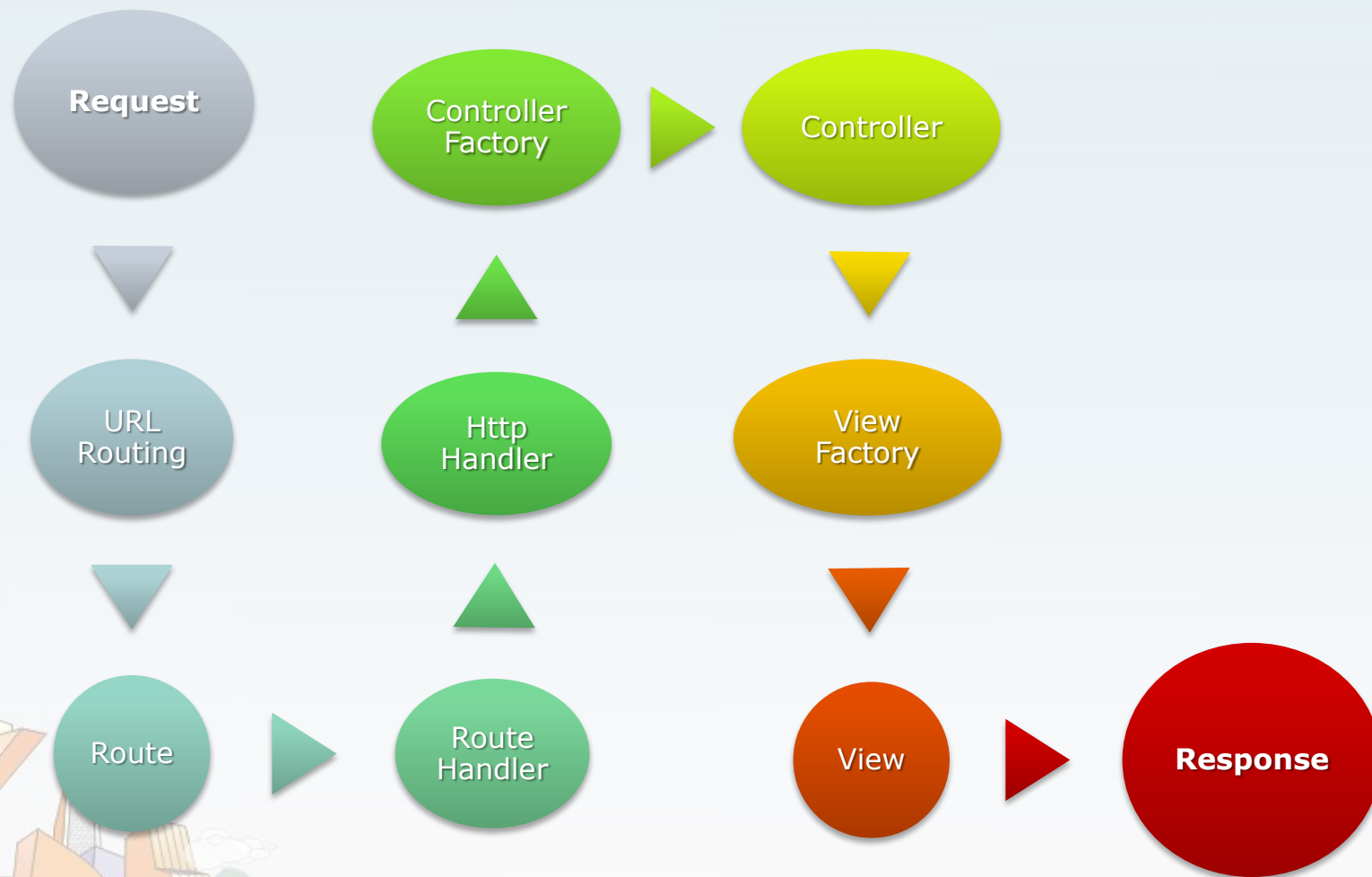
HTTP 回應

- HTTP 重導向(Redirection)



# 瞭解 ASP.NET MVC 執行生命週期

All services from your imperative.



# 聯絡資訊

All services from your imperative.

## ☐ The Will Will Web

記載著 Will 在網路世界的學習心得與技術分享

■ <http://blog.miniasp.com/>

## ☐ Will 保哥的技術交流中心

■ <http://www.facebook.com/will.fans>

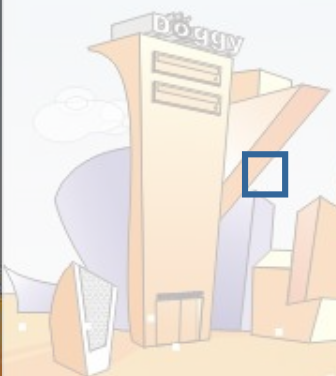


## ☐ ★★★ Will 保哥的噗浪 ★★★

■ <http://www.plurk.com/willh/invite>

## ☐ Will Huang - Google+

■ <http://plus.to/willh>



www.miniasp.com

All services from your imperative.

# 感謝各位

## Q & A

