ASP.NET MVC 4 RC 新增功能介紹

ASP.NET MVC 4 就快出來了，你跟上了嗎？



About Me

- 中文姓名：吳俊毅
- 英文姓名：Gelis Wu
- Email: geliswu@wistronits.com
- MSN：wergelis@msn.com
- Blog：http://www.dotblogs.com.tw/gelis/
- 專長：.NET 相關 Solution、Web 應用、分散式應用程式開發、Architecture Design、系統分析 / 設計。



MVP Adward 2011/2012
Visual C#

ASP.NET MVC 的發展歷史

Date	Version	Notes
2007-12-10	ASP.NET MVC Framework	以CTP方式釋出
2008-03-05	ASP.NET MVC Preview 2	已釋出
2008-05-01	ASP.NET MVC Preview 3	已釋出
2008-07-16	ASP.NET MVC Preview 4	已釋出[7]
2008-08-28	ASP.NET MVC Preview 5	已釋出[8]
2008-10-16	ASP.NET MVC Beta	已釋出[9]
2009-01-27	ASP.NET MVC RC	已釋出[10]
2009-03-03	ASP.NET MVC RC 2	已釋出[11]
2009-03-17	ASP.NET MVC 1.0	已釋出[12]
2009-07-31	ASP.NET MVC 2.0 Preview 1	已釋出[13]
2009-11-17	ASP.NET MVC 2.0 Beta	已釋出[14]
2010-03-11	ASP.NET MVC 2.0 RTM	已釋出[15]
2010-10-06	ASP.NET MVC 3.0 Beta	持續更新中[16]
2010-11-08	ASP.NET MVC 3.0 RC	已釋出[17]
2011-01-13	ASP.NET MVC 3.0 RTM	已釋出[18]
2011-09-20	ASP.NET MVC 4.0 Developer Preview	已釋出[19]
2012-02-15	ASP.NET MVC 4.0 Beta	隨著Microsoft .NET Framework 4.5 RC釋出[20]
2012-05-31	ASP.NET MVC 4.0 RC	[21]

寫在前面

- 雖然MVC 4 有些新增功能到了RC的時候被移除掉了，不過它還是可圈可點之處啦。
- 感覺 MVC 4 RTM 好像拖了很久！不過開發人員是辛苦的，所以我們也不要太計較。
- 雖然讓我們等了很久，不過我們相信，我們的等待是值得的
- RTM正式版已於2012/8/15上市了

大綱

- 所使用的開發環境
- 如何將 MVC3 的專案升級成 MVC4
- ASP.NET MVC 4 RC 新增功能介紹

大綱

- 所使用的開發環境
- 如何將 MVC3 的專案升級成 MVC4
- ASP.NET MVC 4 RC 新增功能介紹

所使用的開發環境

- Windows 7 Home Premium
- Visual Studio 2010 Ultimate &.NET Framework 4.0
- Visual Studio 2012 Ultimate RC & .NET Framework 4.5 RC
- 在Visual Studio 2012 RC 中使用的版本已經是MVC 4.0 RC的版本
- 建議使用 Windows Platform Installer 4.0 來安裝。
- 當然也可以下載Visual Studio 2010使用的 MVC 4.0 RC 版本
- 或者使用8/15最新Release的Visual Studio 2012 RTM

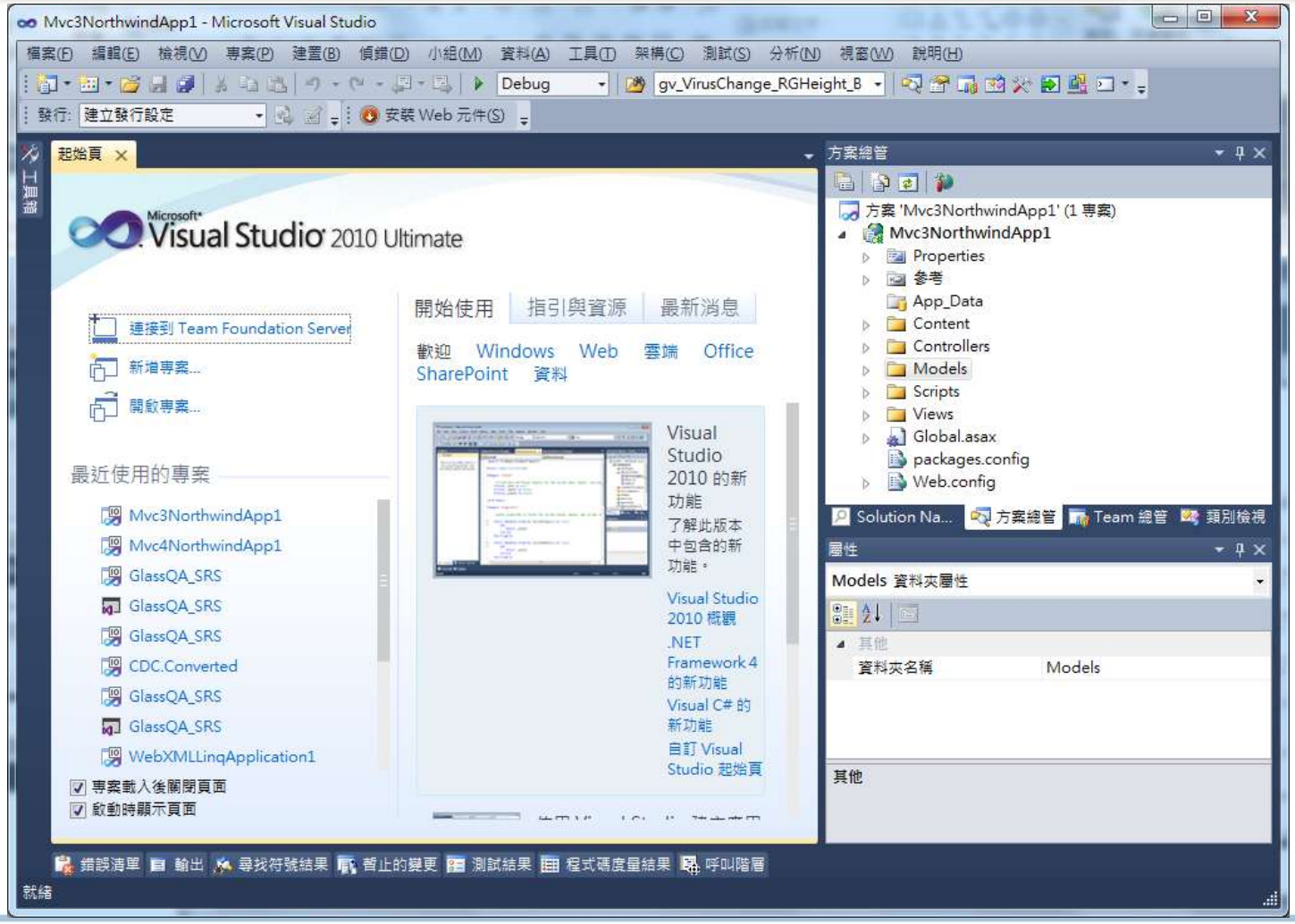
大綱

- 所使用的開發環境
- 如何將 MVC3 的專案升級成 MVC4
- ASP.NET MVC 4 RC 新增功能介紹

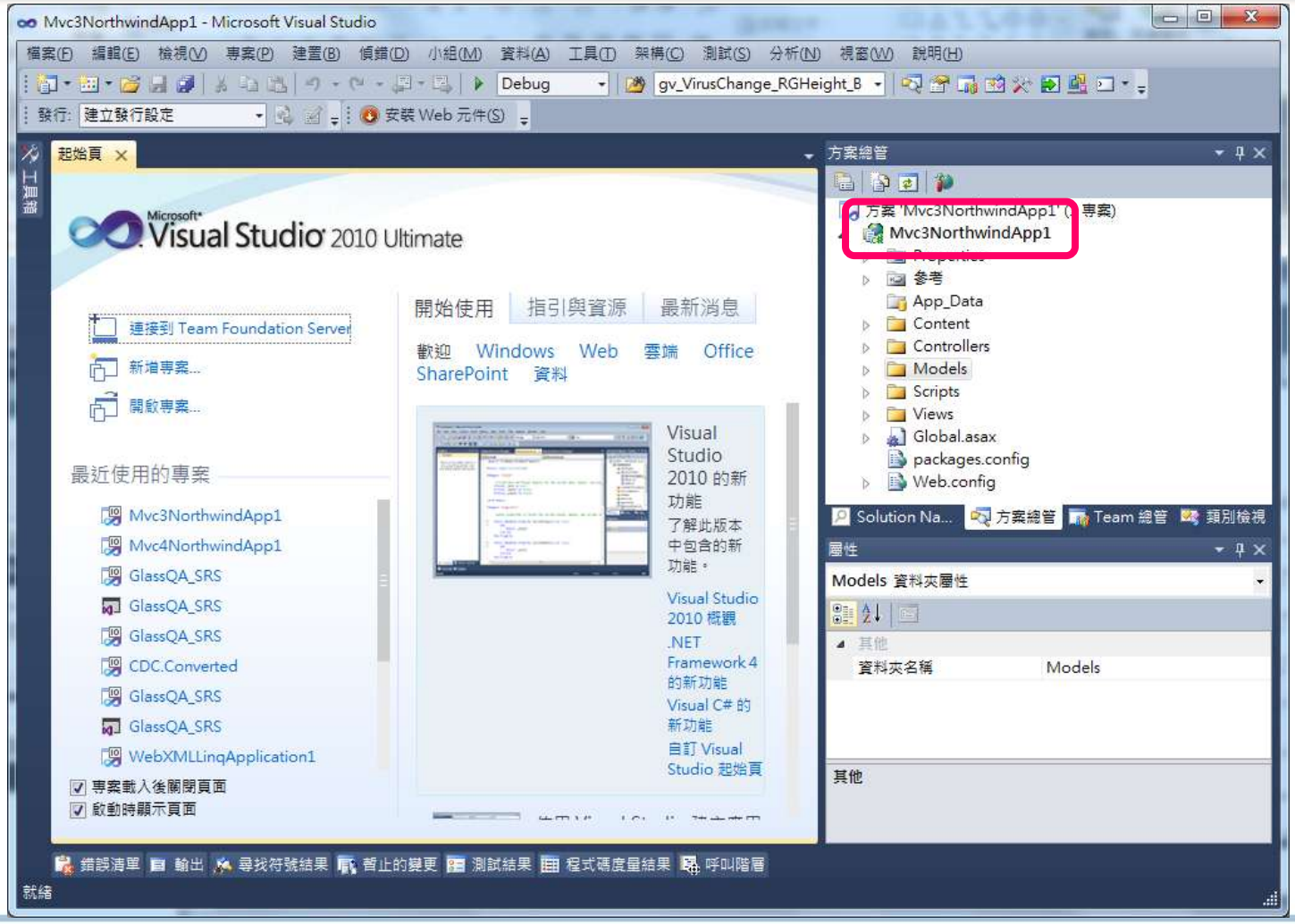
第一式、乾坤大挪移

- 要將 ASP.NET MVC3 轉成 MVC4 絕招、第一式、乾坤大挪移。
- 所謂的乾坤大挪移就是➔ COPY
- 因此我們首先得建立一個空的ASP.NET MVC 4.0 的專案

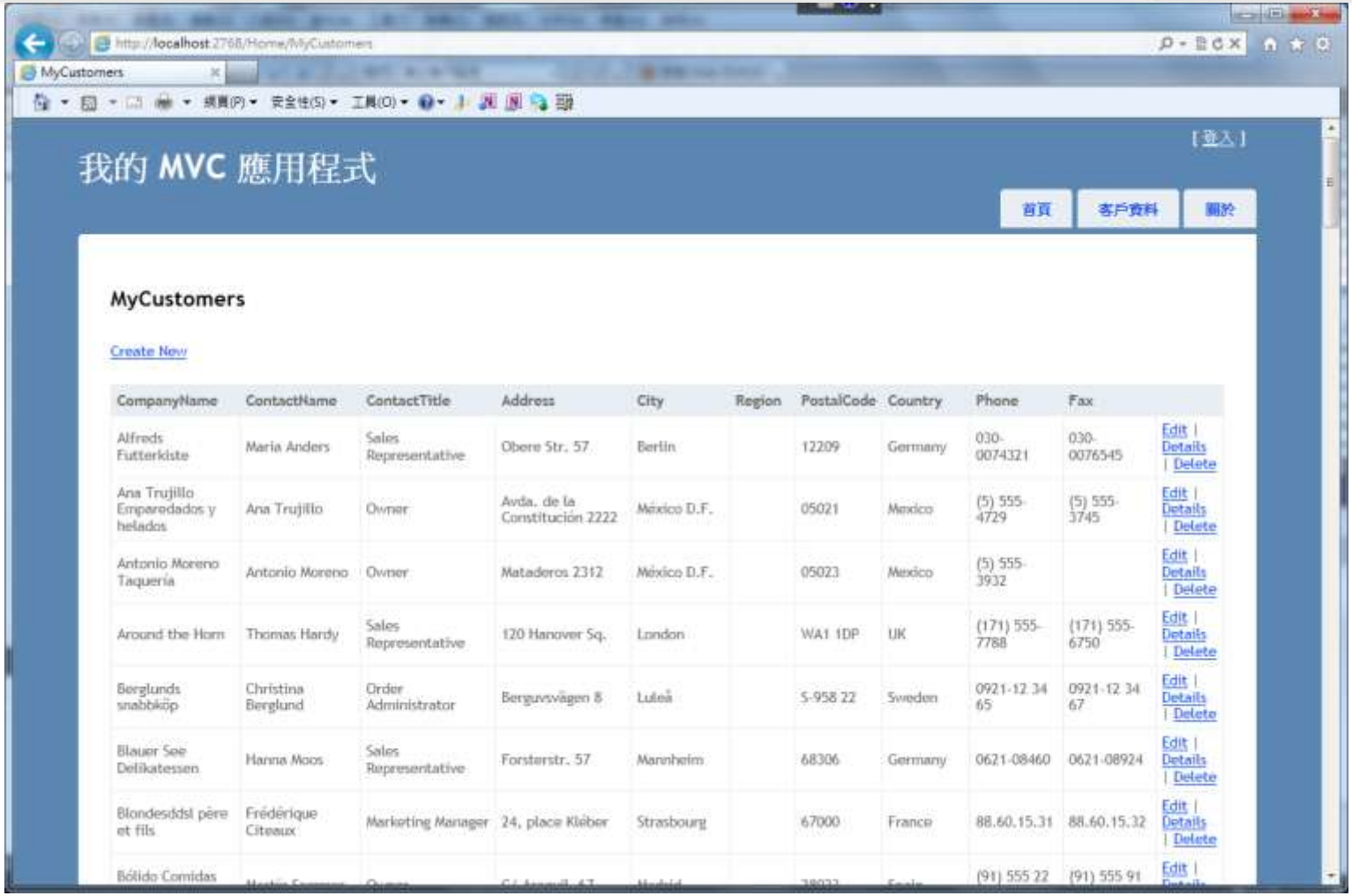
Demo、使用現有MVC3專案



Demo、使用現有MVC3專案

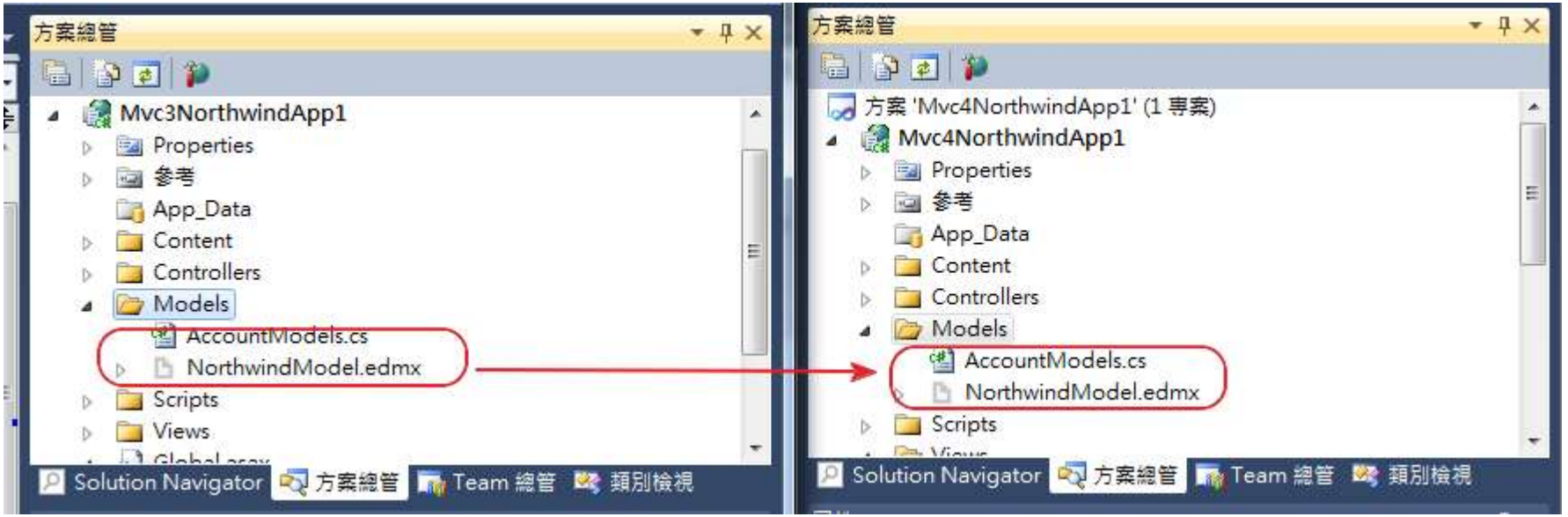


Demo、使用現有MVC3專案



步驟一、先搬Model

- 首先先建立好一個空的ASP.NET MVC 4.0的應用程式。



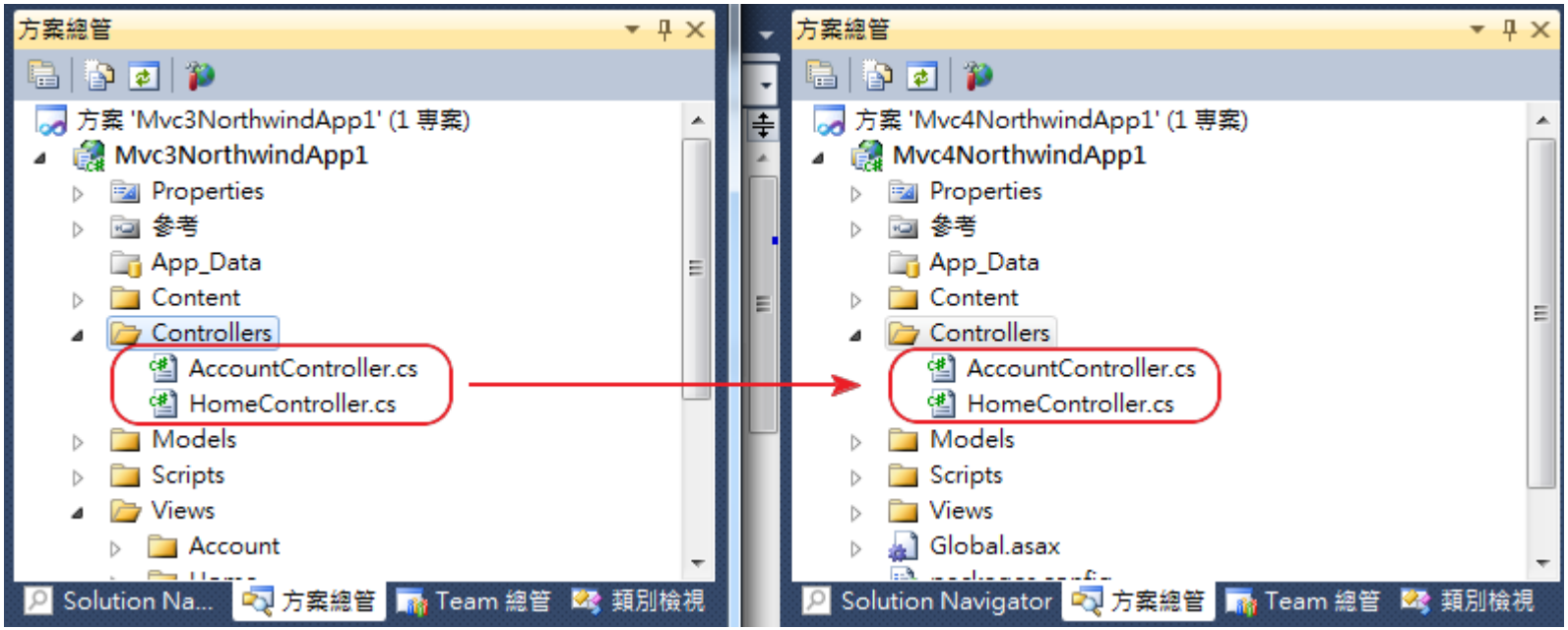
步驟一、先搬Model

- 首先先建立好一個空的ASP.NET MVC 4.0的應用程式。

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.Globalization;
5  using System.Web.Mvc;
6  using System.Web.Security;
7
8  namespace Mvc4NorthwindApp1.Models
9  {
10
11     public class ChangePasswordModel
12     {
13         [Required]
14         [DataType(DataType.Password)]
15         [Display(Name = "目前密碼")]
16         public string OldPassword { get; set; }
17
18         [Required]
19         [StringLength(100, ErrorMessage = "{0} 長度至少必須為 {2} 個字
20         [DataType(DataType.Password)]
```

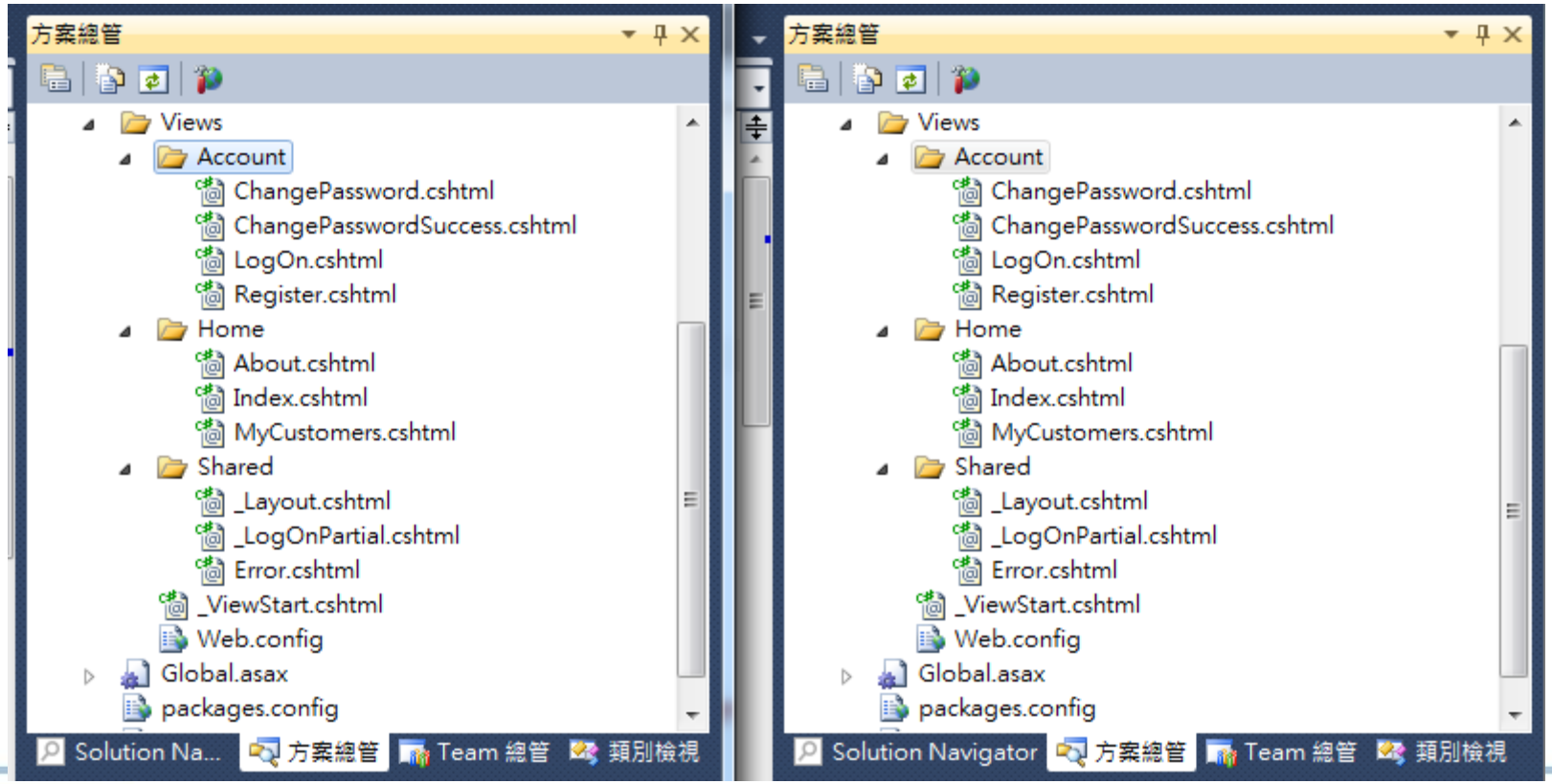
如果Namespace有變動，記得要修改。

步驟二、再搬Controller



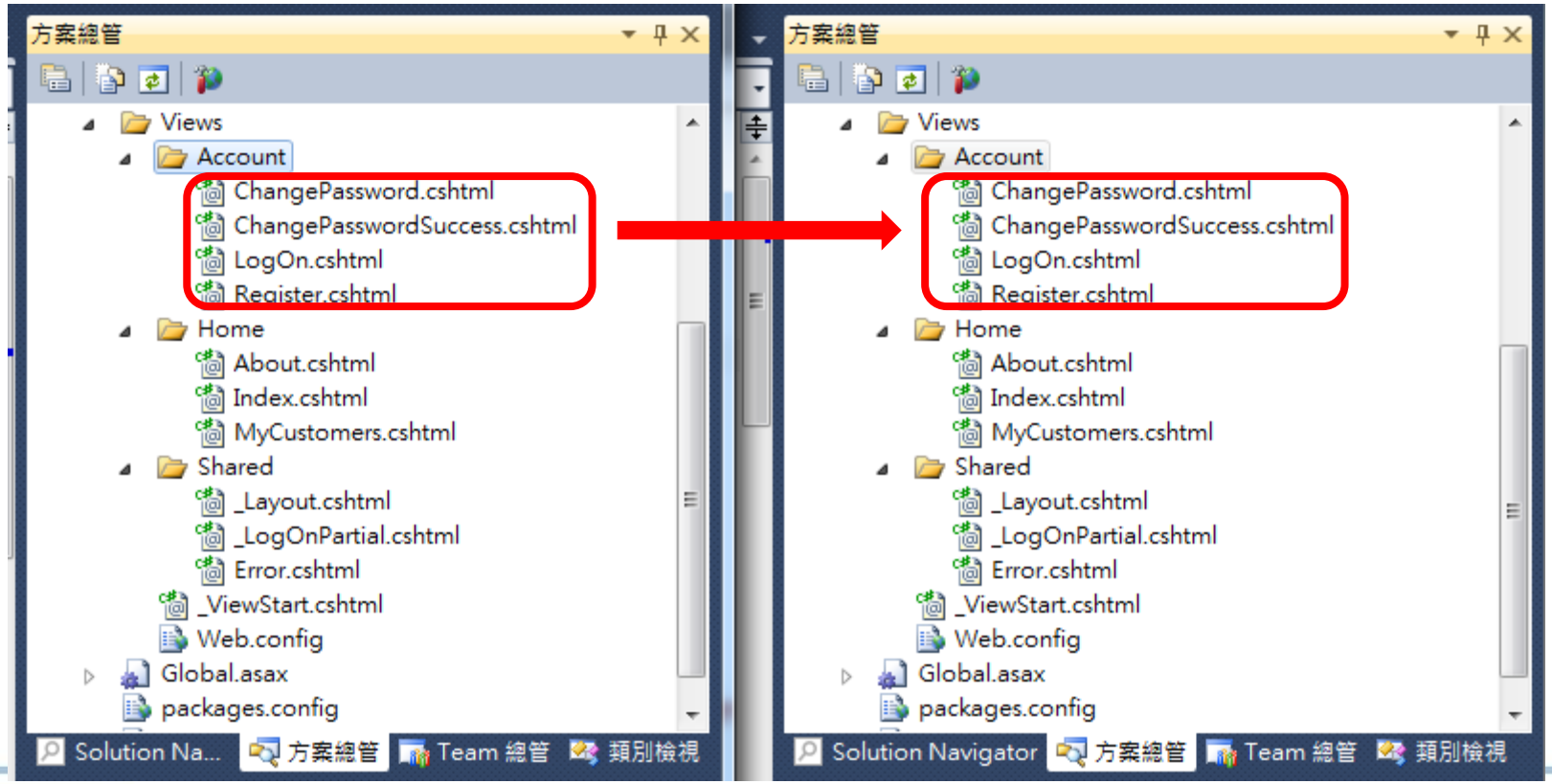
步驟三、續搬View

- 不是所有都一起搬過去。



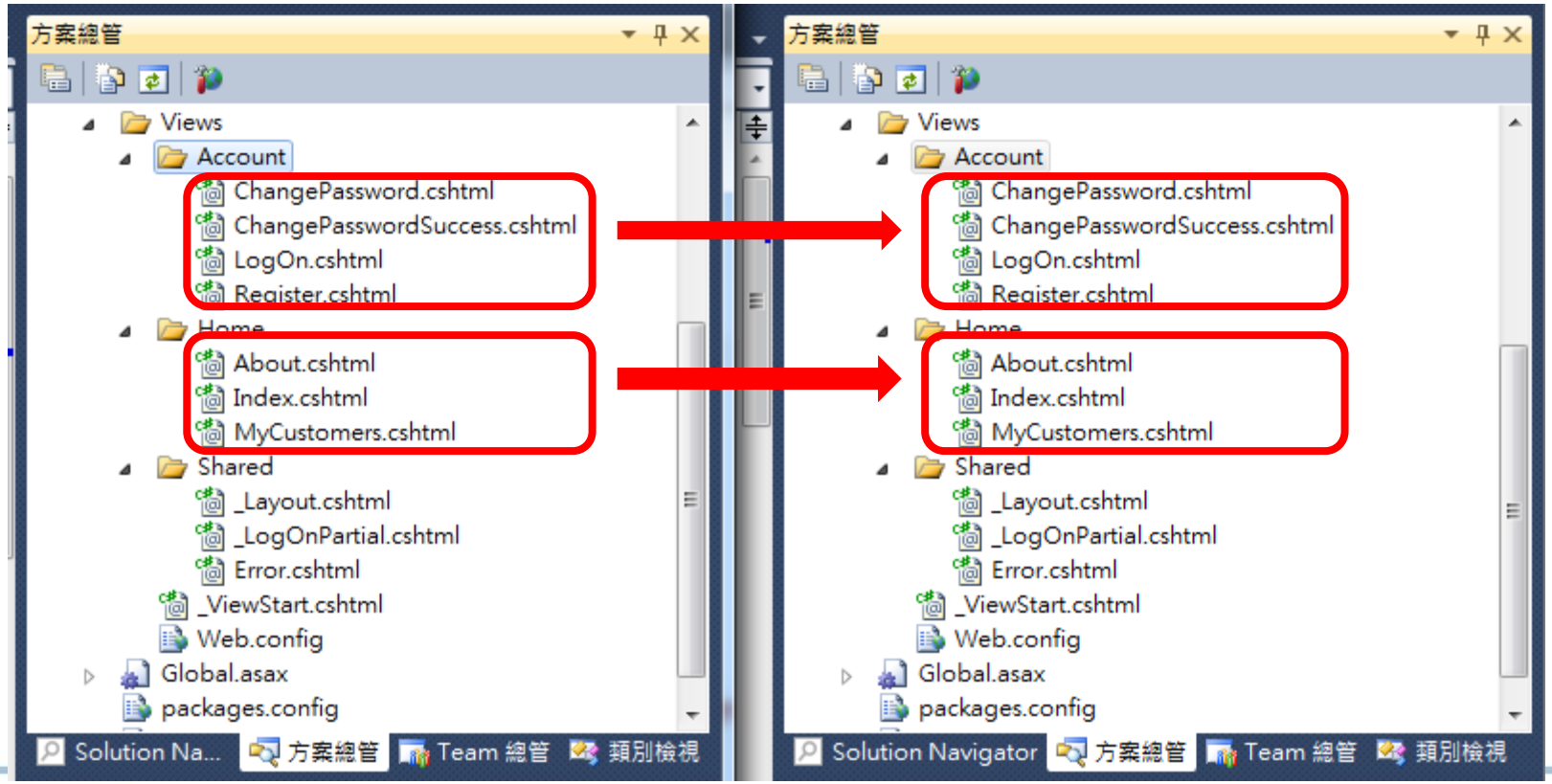
步驟三、續搬View

- 不是所有都一起搬過去。



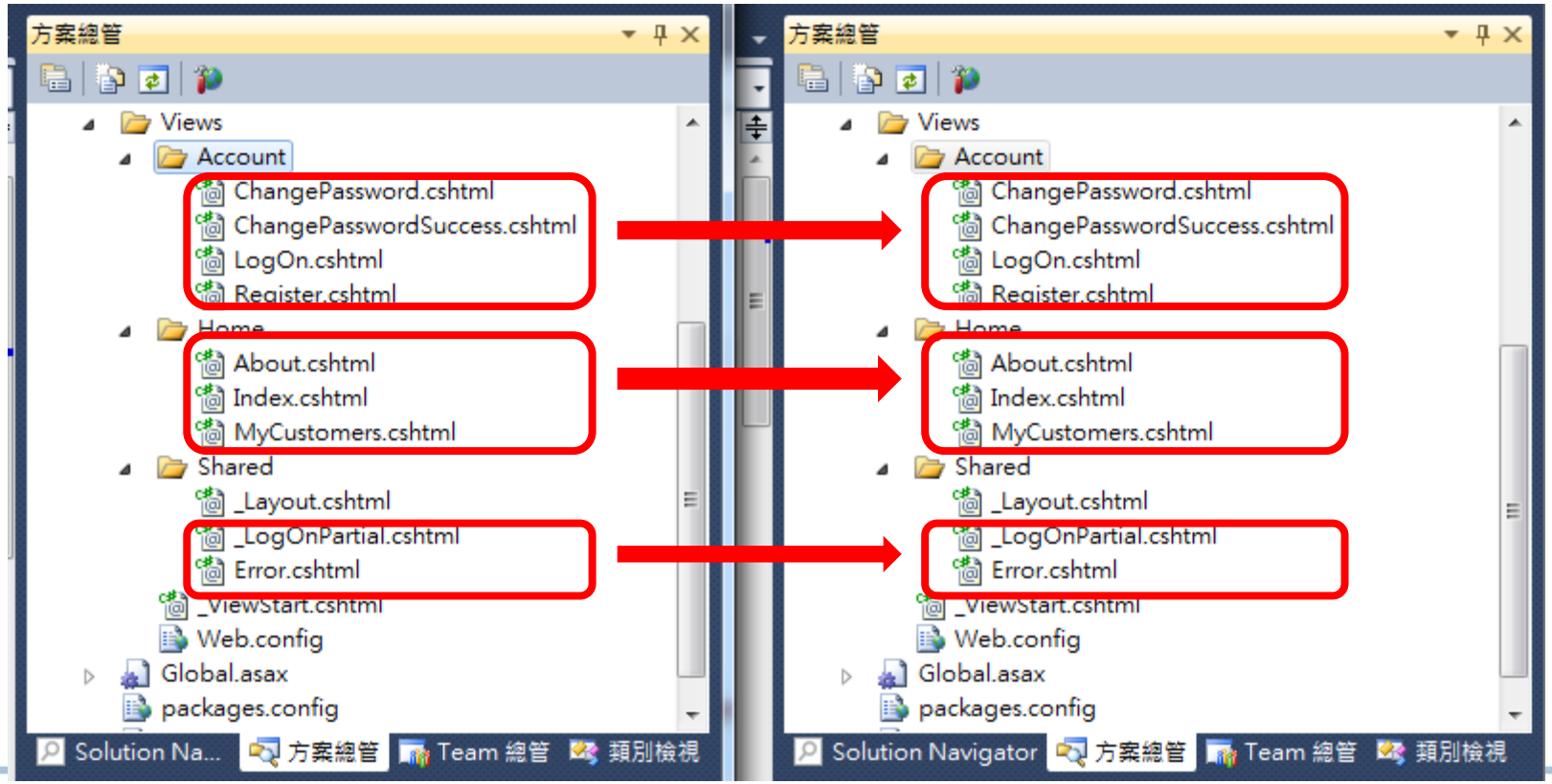
步驟三、續搬View

- 不是所有都一起搬過去。



步驟三、續搬View

- 不是所有都一起搬過去。



步驟三、續搬View

- 不是所有都一起搬過去。

```
1  @model Mvc4NorthwindApp1.Models.RegisterModel
2
3  @{
4      ViewBag.Title = "註冊
5  }
6
7  <h2>建立新帳戶</h2>
8  <p>
9      使用下面的表單建立
10 </p>
11 <p>
12     密碼的長度至少必須為 @Membership.MinRequiredPasswordLength 個字元。
13 </p>
14
15 <script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="t
16 <script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.j
17
18 @using (Html.BeginForm()) {
19     @Html.ValidationSummary(true, "帳戶建立失敗。請更正錯誤後再試一次。
```

記得修改model的Namespace

步驟三、續搬View(2)

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    <link href="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/Content/css")" rel="stylesheet" type="text/css" />
    <link href="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/Content/themes/base/css")" rel="stylesheet" type="text/css'
    <script src="@System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl("~/Scripts/js")"></script>
</head>
<body>
    <div class="page">
        <header>
            <div id="title">
                <h1>我的 MVC 應用程式</h1>
            </div>
            <div id="logindisplay">
                @Html.Partial("_LogOnPartial")
            </div>
            <nav>
                <ul id="menu">
                    <li>@Html.ActionLink("首頁", "Index", "Home")</li>
                    <li>@Html.ActionLink("客戶資料", "MyCustomers", "Home")</li>
                    <li>@Html.ActionLink("關於", "About", "Home")</li>
                </ul>
            </nav>
        </header>
        <section id="main">
            @RenderBody()
        </section>
        <footer>
        </footer>
    </div>
</body>
</html>
```

步驟四、確認 web.config

- MVC 4 中新增兩個 appSeettings 的屬性值

```
<appSettings>
  <add key="webpages:Version" value="2.0.0.0" />
  <add key="PreserveLoginUrl" value="true" />
</appSettings>
```

- MVC 3 相關參考版本

```
System.Web.Mvc, Version=3.0.0.0
System.Web.WebPages, Version=1.0.0.0
System.Web.Helpers, Version=1.0.0.0
System.Web.WebPages.Razor, Version=1.0.0.0
```

- MVC 4 相關參考版本

```
System.Web.Mvc, Version=4.0.0.0
System.Web.WebPages, Version=2.0.0.0
System.Web.Helpers, Version=2.0.0.0,
 System.Web.WebPages.Razor, Version=2.0.0.0,
```

步驟五、確認 web.config（續）

- 如果你的MVC應用程式有使用到一些third-party的Library，如果它可能使用到舊版的mvc相關參考，請再<configuration>加入如下敘述：

```
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-2.0.0.0" newVersion="2.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-4.0.0.0" newVersion="4.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35" />
      <bindingRedirect oldVersion="1.0.0.0-2.0.0.0" newVersion="2.0.0.0" />
    </dependentAssembly>
  </assemblyBinding>
</runtime>
```

大綱

- 所使用的開發環境
- 如何將 MVC3 的專案升級成 MVC4
- ASP.NET MVC 4 RC 新增功能介紹

ASP.NET MVC 4 RC 新增功能介紹

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

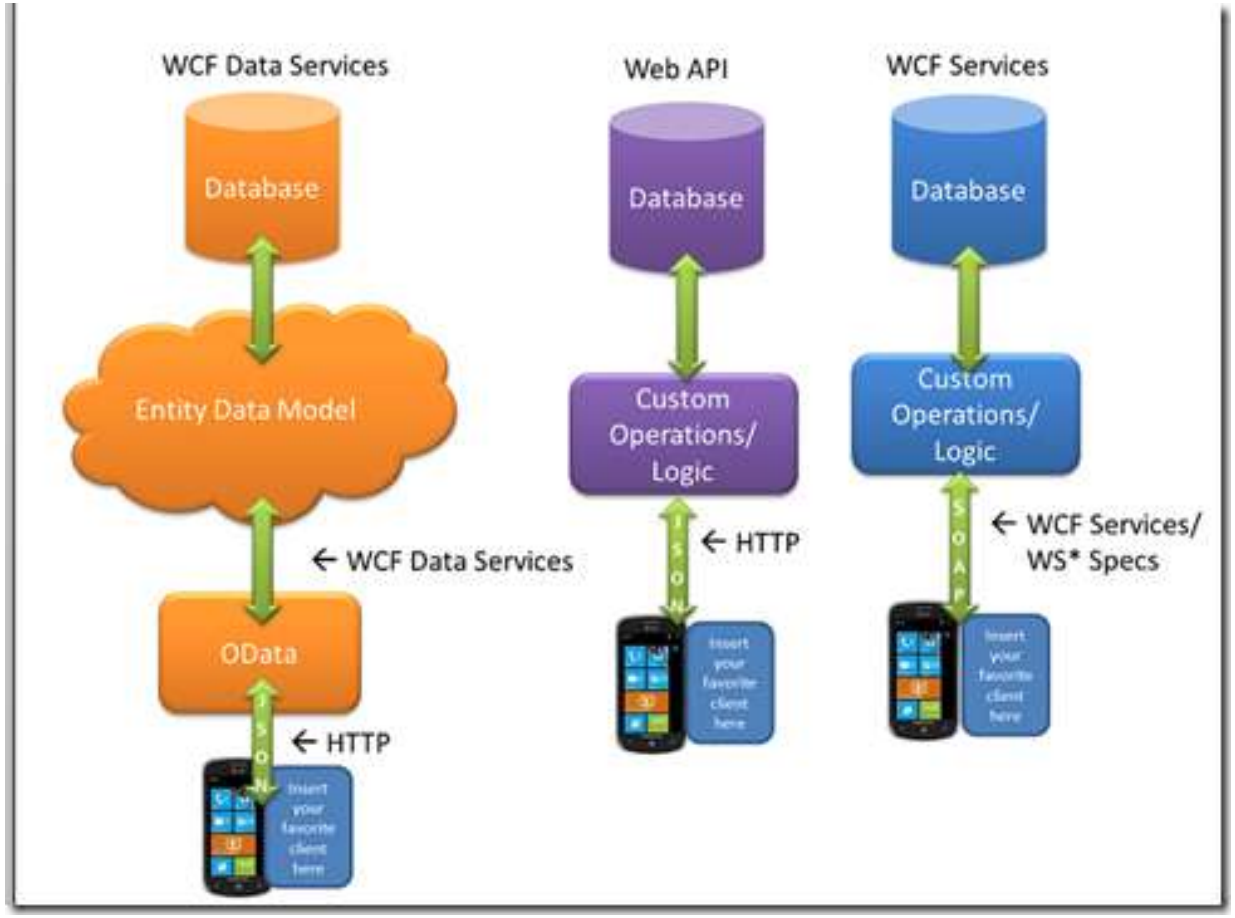
ASP.NET MVC 4 RC 新增功能介紹

- **ASP.NET Web API**
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

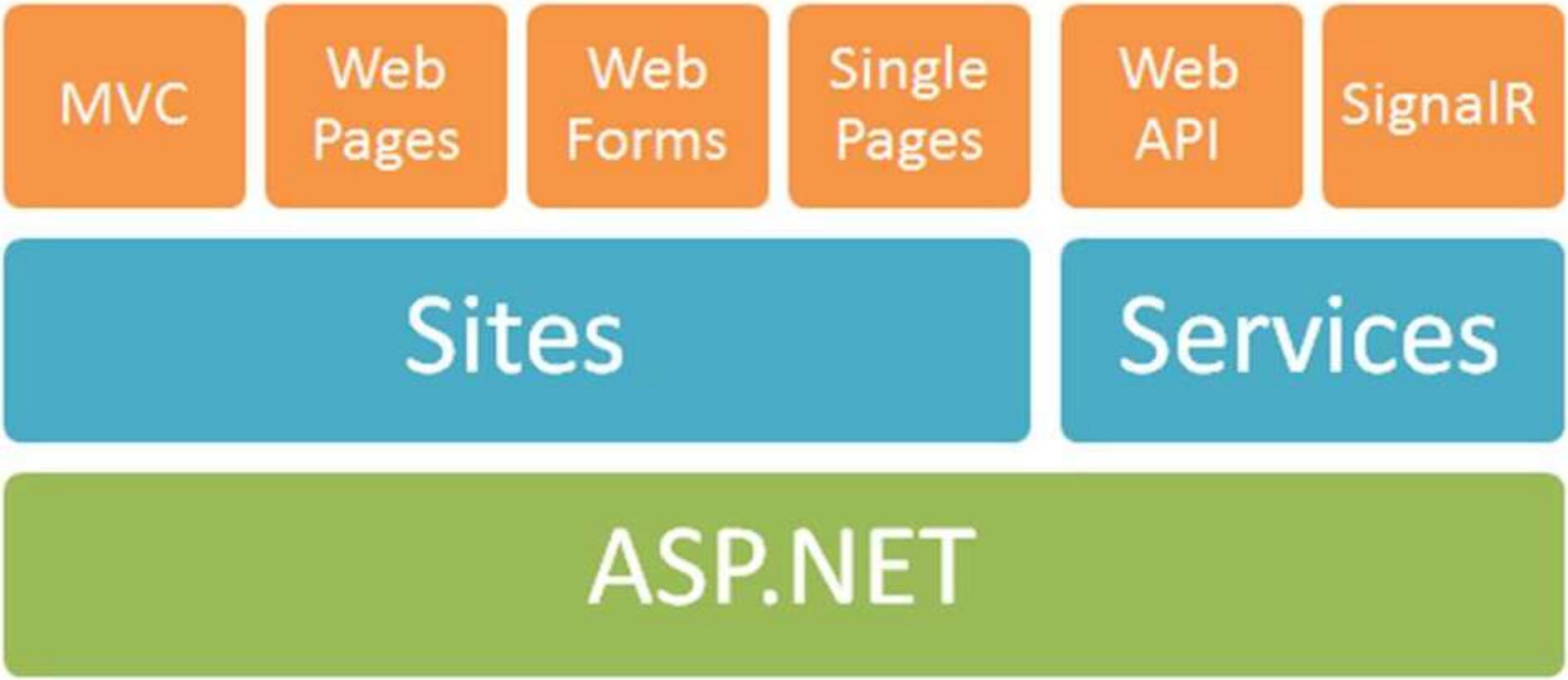
認識 ASP.NET Web API

- 同 WCF Data Service ，且支援RESTful介面，但是它更簡化，更簡單，易於實作。
- 只支援HTTP協定，透過 GET/POST 方式提供遠端使用XML或JSON資料格式進行呼叫與使用。
- 更容易讓多種裝置使用：如平板、智慧型裝置、桌上型電腦、其他平台服務、瀏覽器…
- 支援非同步執行特性，有更加之延展性
- 支援 ASP.NET MVC 的 Routing 功能
- 強行別的HttpRequestMessage與HttpResponseMessage，因此模型繫結與驗證更加容易。

Web API vs. WCF Service vs. WCF Data Service



ASP.NET 4.5 Web API 架構



Demo 1

- 使用 Visual Studio 2012 RC 的 Web API 樣板建立 Web API 應用程式。
- 使用 jQuery 取得資料。

Demo 2

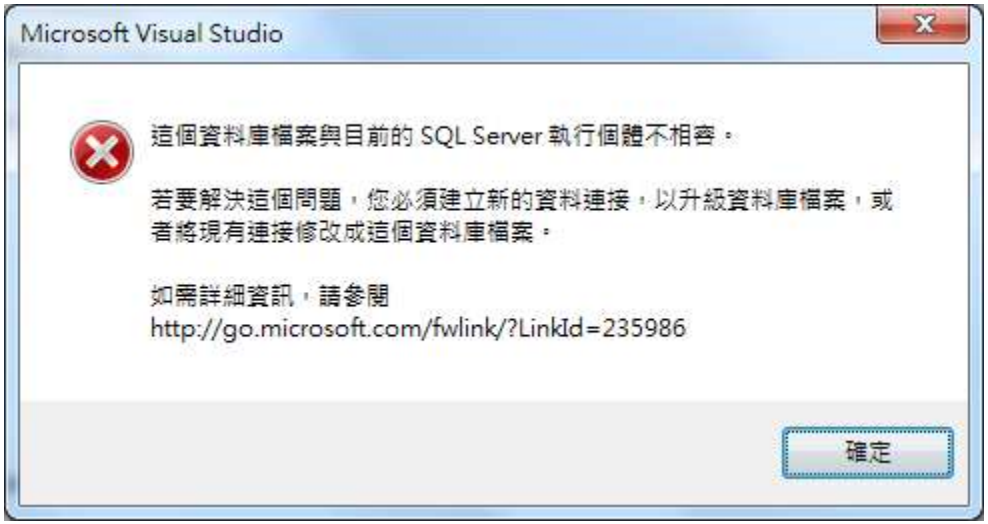
- 範例實作、從無到有，建立一個 CRUD 應用程式

Demo 2（續）

- 使用 SQL Express
- Vistual Studio 2012 RC 會提示資料庫不相容，手動升級即可。

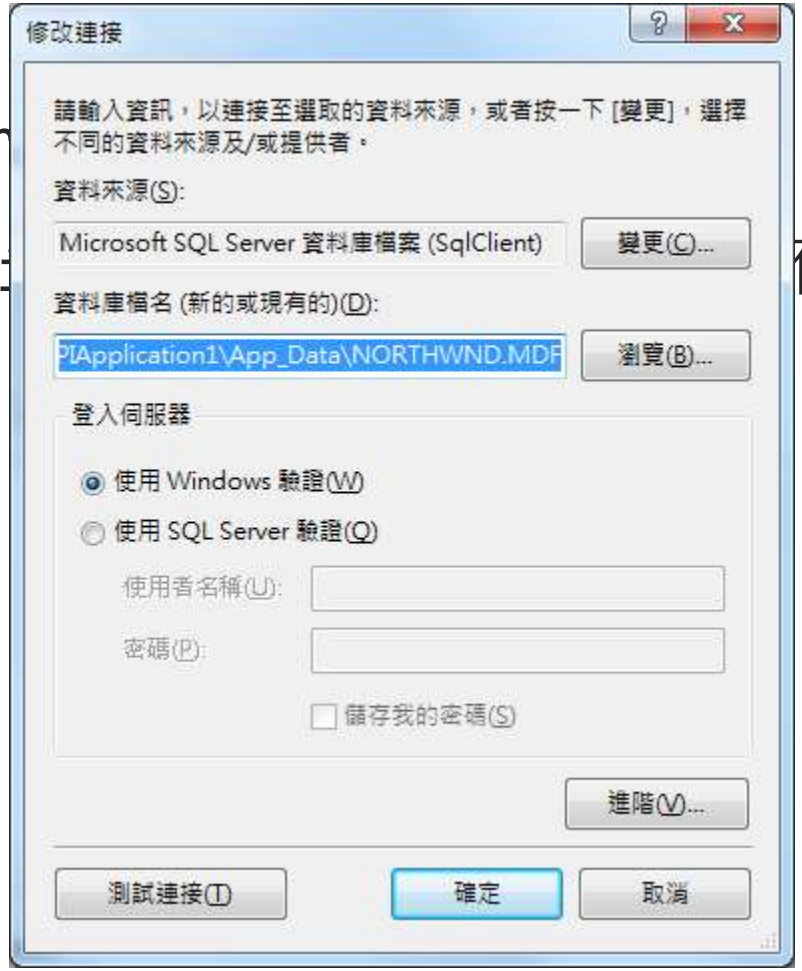
Demo 2（續）

- 使用 SQL Express
- Vistual Studio 2012 RC 會提示資料庫不相容，手動升級即可。



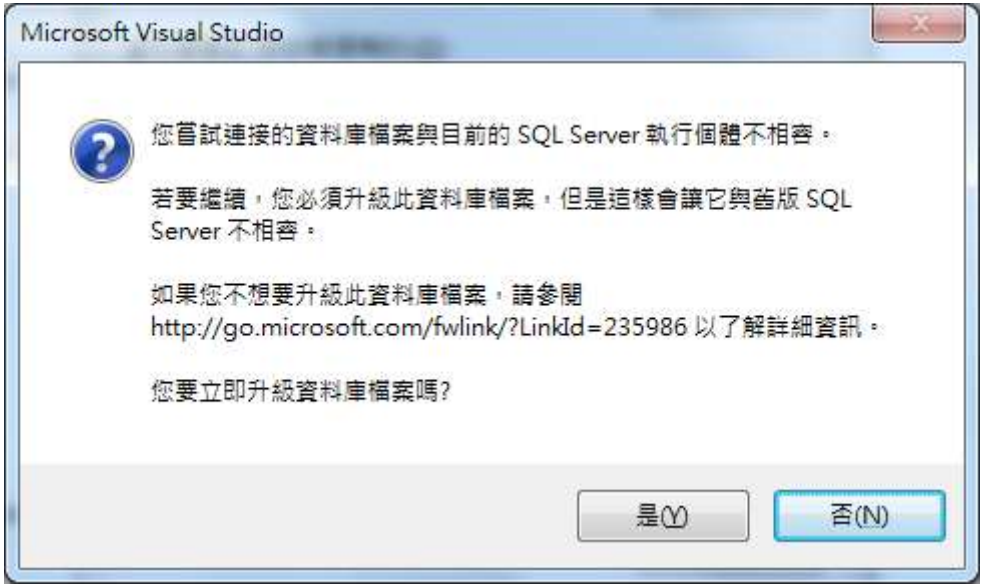
Demo 2（續）

- 使用 SQL Expr
- Vistual Studi 不相容，手動升級即可。



Demo 2 (續)

- 使用 SQL Express
- Vistual Studio 2012 RC 會提示資料庫不相容，手動升級即可。



Self-Host 掛載 Web API (1)

- 當你沒有引用 JSON.NET 物件時，無法掛載應用程式

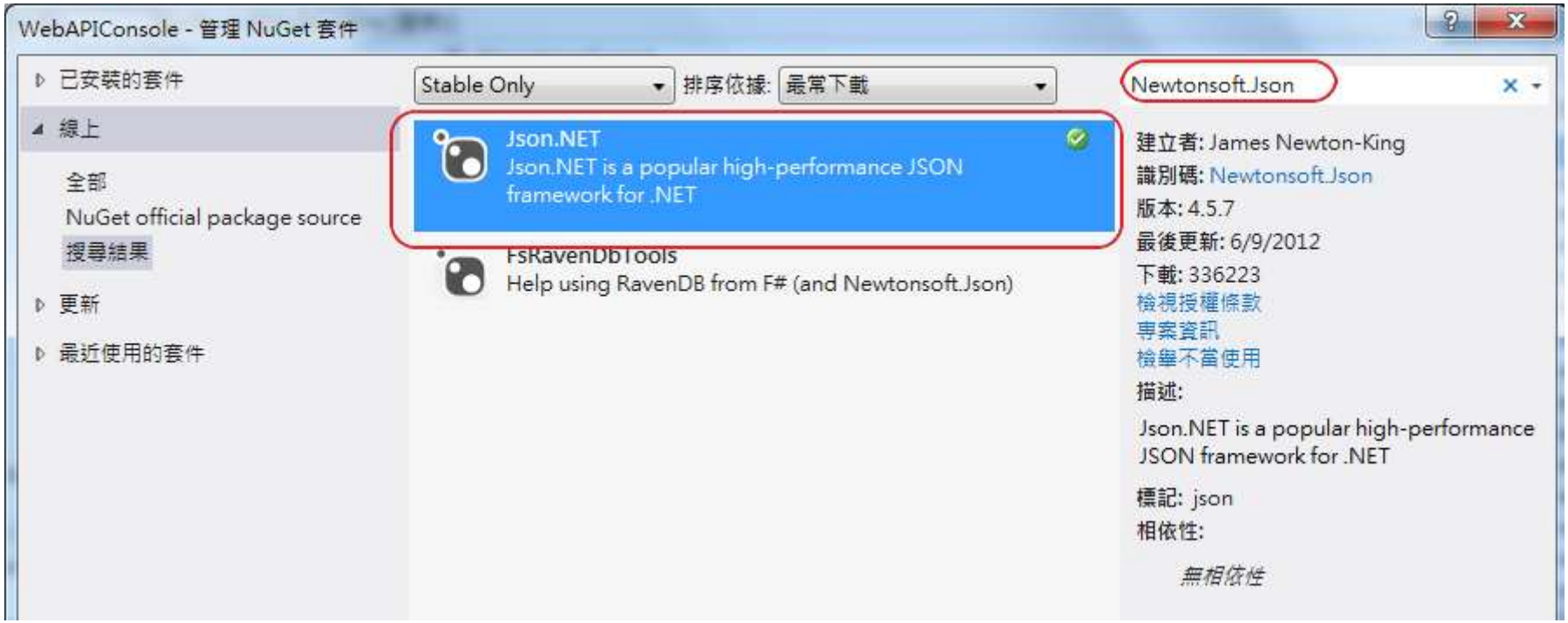
```
var config = new HttpSelfHostConfiguration("http://localhost:8000");
config.Routes.MapHttpRoute("MyCusAPI", "api/{controller}/id", new { id = RouteParameter.Optional });

HttpSelfHostServer server = new HttpSelfHostServer(config);
server.OpenAsync().Wait();
Console.ReadLine();
```



Self-Host 掛載 Web API (1)

- 當你沒有引用 JSON.NET 物件時，無法掛載應用程式



Self-Host 掛載 Web API (2)

- 必須以系統管理員權限執行，否則……

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Http;
using System.Web.Http.SelfHost;

namespace WebAPIConsole
{
    class Program
    {
        static void Main(string[] args)
        {
            var config = new HttpSelfHo
            config.Routes.MapHttpRoute(

            HttpSelfHostServer server =
            server.OpenAsync().Wait();
            Console.ReadLine();

        }
    }
}
```

檢視詳細資料

例外狀況快照(E):

System.AggregateException	Count = 1
[System.AggregateException]	Count = 1
Data	{System.Collections.ListDictionaryInternal}
HelpLink	null
HResult	-2146233088
InnerException	{"HTTP 無法登錄 URL http://+:8000/。您的處理程序沒有足夠的存取權可存取此命名空間 (如需詳細資訊，請參閱 http://
InnerExceptions	Count = 1
Message	發生一或多項錯誤。
Source	mscorlib
StackTrace	於 System.Threading.Tasks.Task.ThrowIfExceptional(Boolean includeTaskCanceledExceptions) 於 System.Threa
TargetSite	{Void ThrowIfExceptional(Boolean)}

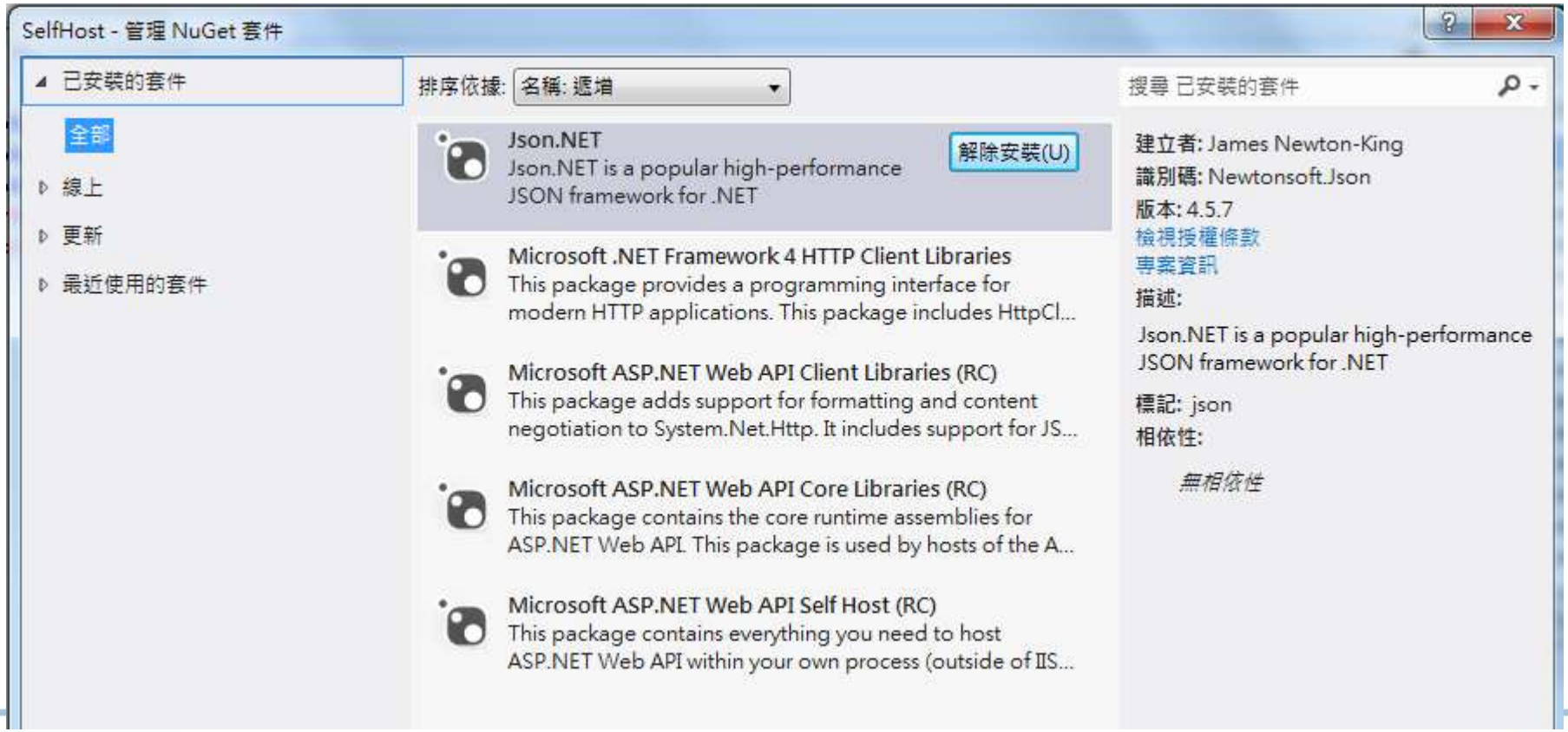
確定

Self-Host 掛載 Web API (3)

- 將 EDM Model 包裝成 Class Library
- 建立 Self-Host Console 專案
- 將 MyCusAPIController.cs 檔案加入Self-Host Console 專案中

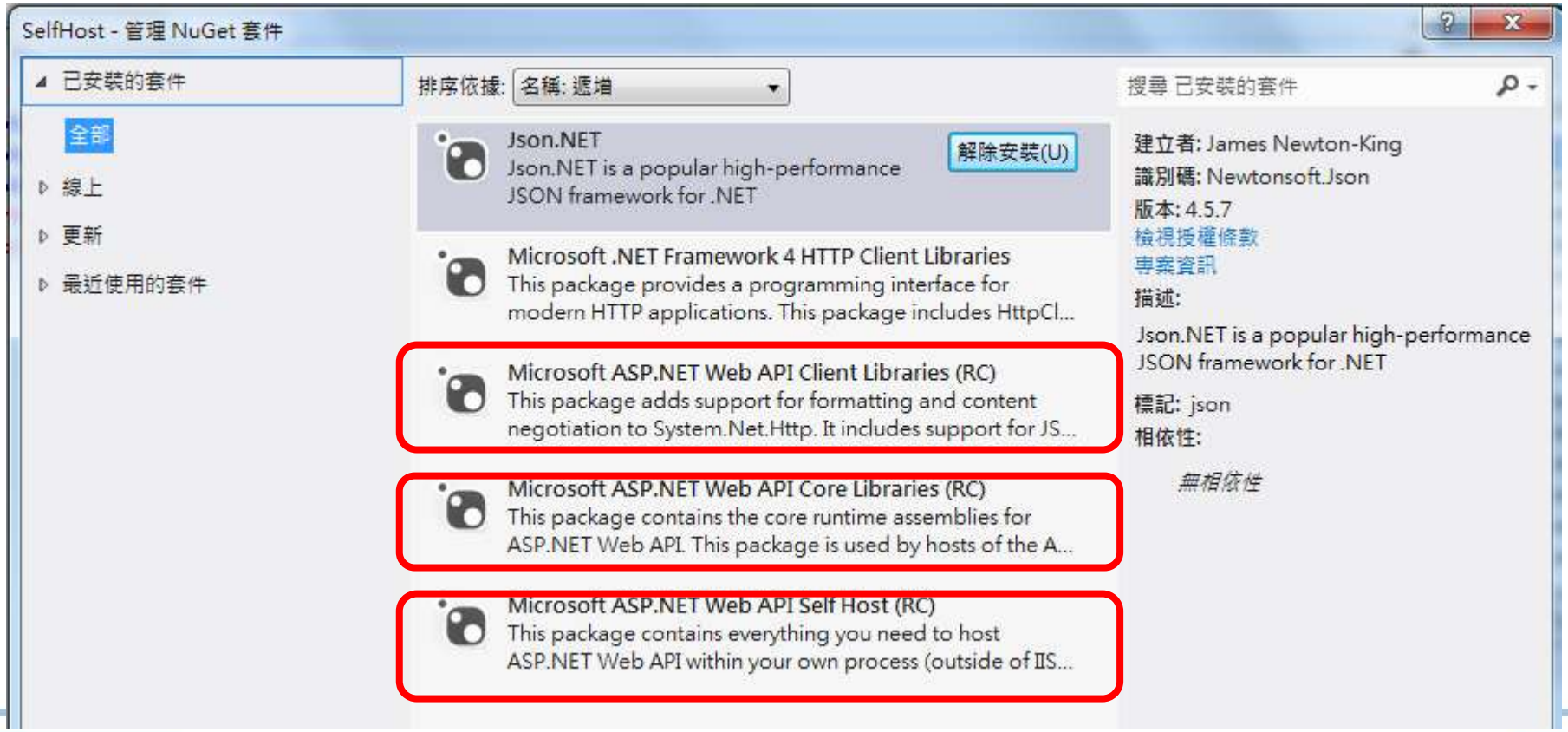
Self-Host 掛載 Web API (4)

- Seft-Host 基本必須安裝的套件



Self-Host 掛載 Web API (4)

- Seft-Host 基本必須安裝的套件



Self-Host 掛載 Web API (5)

- 撰寫 Self-Host 掛載程式碼

```
class Program
{
    static void Main(string[] args)
    {
        var config = new HttpSelfHostConfiguration("http://localhost:8080");
        config.Routes.MapHttpRoute(
            name: "DefaultApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );

        HttpSelfHostServer server = new HttpSelfHostServer(config);
        server.OpenAsync().Wait();
        Console.WriteLine("The server is running...");
        Console.ReadLine();

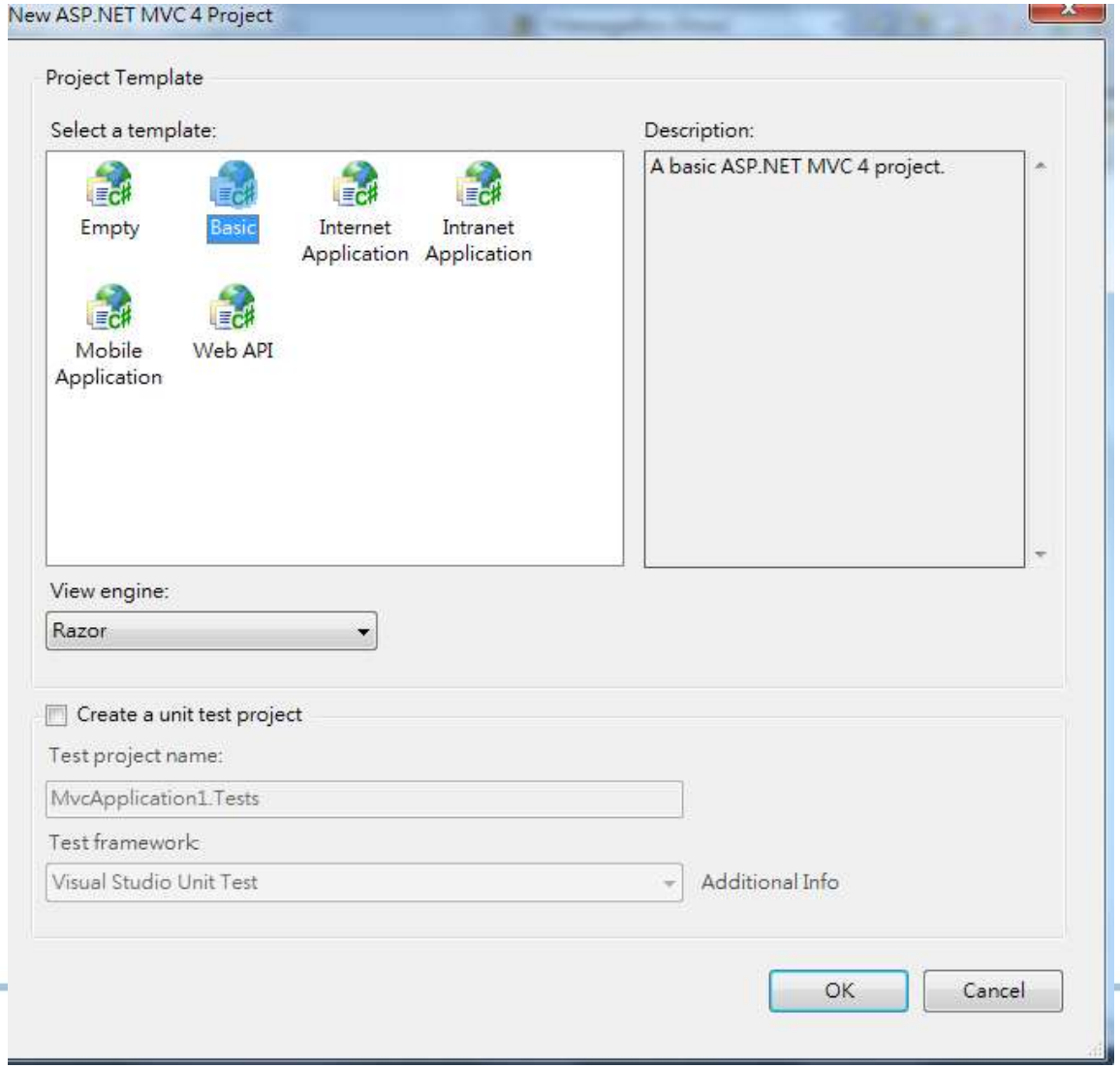
    }
}
}
```

Self-Host 掛載 Web API (6)

- Demo…

Enhancements to Default Project Templates

- ASP.NET Web API
- **Enhancements to Default Project Templates**
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

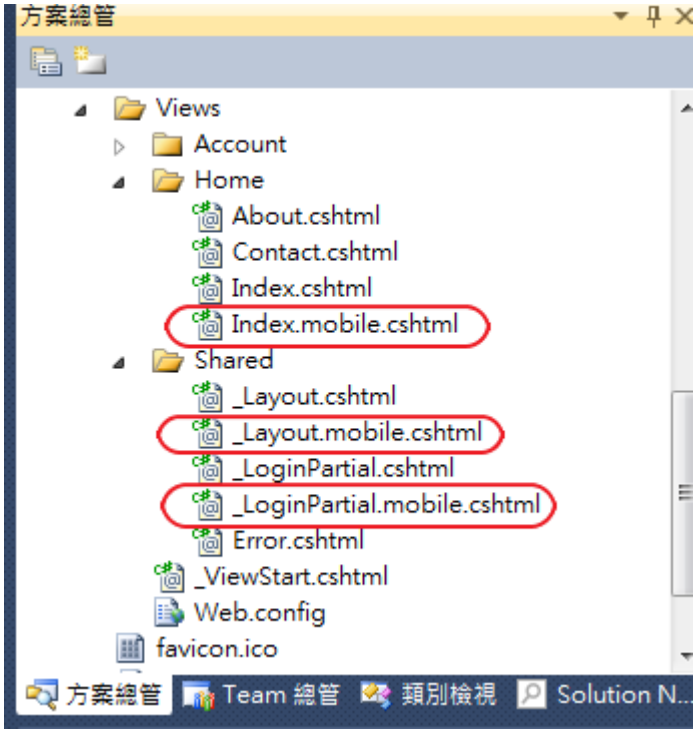

New ASP.NET MVC 4 Project
Project Template
Select a template:
Empty
Basic
Internet Application
Intranet Application
Mobile Application
Web API
Description:
A basic ASP.NET MVC 4 project.
View engine:
Razor
Create a unit test project
Test project name:
MvcApplication1.Tests
Test framework
Visual Studio Unit Test
Additional Info
OK
Cancel

- ASP.NET Web API
- Enhancements to Default Project Templates
- **Mobile Project Template**
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

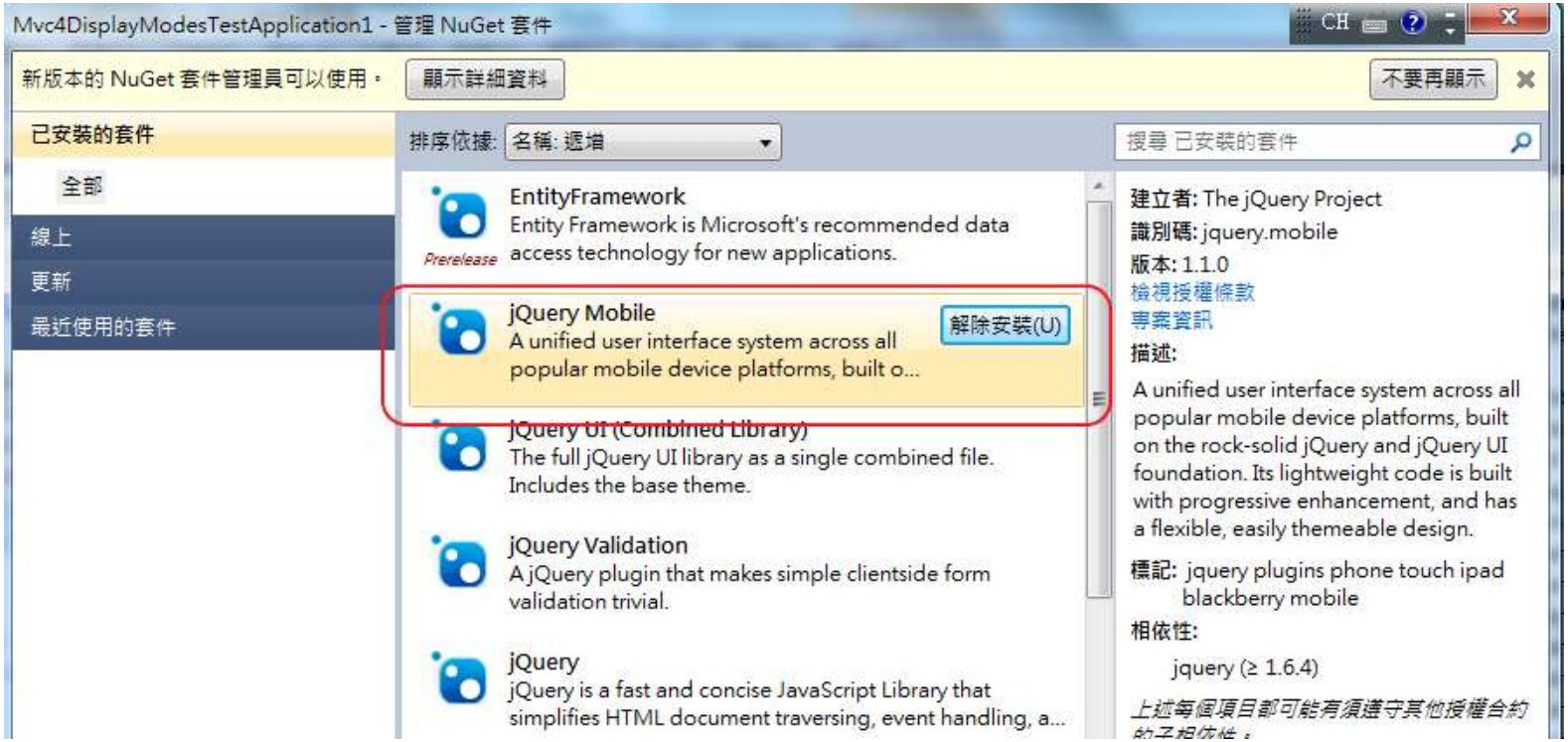
Display Modes

- 判別是否使用手持智慧裝置上網
- 使用 檔名 來辨別要顯示的檔案
- 從mobile的MVC專案複製相關的檔案，並將檔案名稱修改為 XXX.mobile.cshtml
- mobile的jQuery相關檔案也必須複製過來。



Display Modes (2)

- 屬於 mobile 使用的 jQuery 檔案可透過 NeGet 來安裝



Display Modes (3)

- 若您還要區分不同的掌上型裝置檢視不同的View時，可以在 Global.asax.cs 加入如下敘述

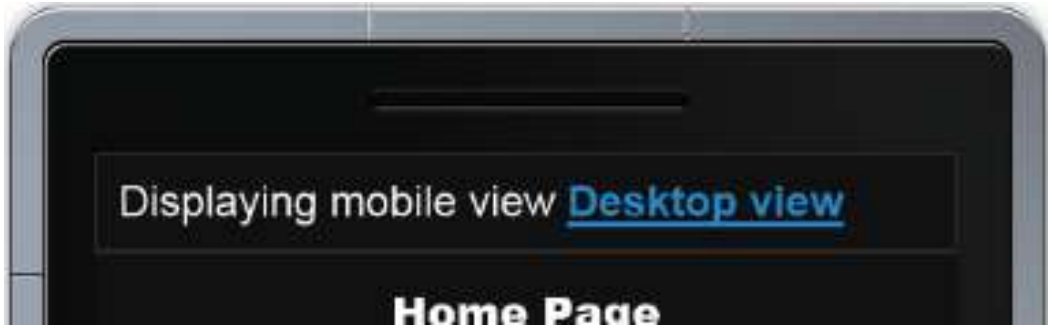
```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);

        DisplayModeProvider.Instance.Modes.Insert(0, new DefaultDisplayMode("iPhone")
        {
            ContextCondition = (context => context.GetOverriddenUserAgent().IndexOf
                ("iPhone", StringComparison.OrdinalIgnoreCase) >= 0)
        });
    }
}
```

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- **jQuery Mobile, the View Switcher, and Browser Overriding**
- Task Support for Asynchronous Controllers
- Azure SDK
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

jQuery Mobile, the View Switcher, and Browser Overriding



- 讓掌上型智慧裝置使用者也可以選擇使用PC版本檢視網頁
- 目前Yahoo、MSN 等大型入口網站皆提供此功能。
- 透過 User Agent 判斷，因此可以做到特定平台使用特定的 View
- 通常與 Display Modes 一起使用。

jQuery Mobile, the View Switcher, and Browser Overriding (2)

- 使用方式，在Views/Shared的底下放置

  _ViewSwitcher.cshtml檔案，並撰寫如下內容：

```
@if (Request.Browser.IsMobileDevice && Request.HttpMethod == "GET")
{
    if (ViewContext.HttpContext.GetOverriddenBrowser().IsMobileDevice)
    {
        @Html.ActionLink("Desktop view", "SwitchView", "ViewSwitcher",
            new { mobile = false, returnUrl = Request.Url.PathAndQuery },
            new { rel = "external" })
    }
    else
    {
        @Html.ActionLink("Mobile view", "SwitchView", "ViewSwitcher",
            new { mobile = true, returnUrl = Request.Url.PathAndQuery },
            new { rel = "external" })
    }
}
```

jQuery Mobile, the View Switcher, and Browser Overriding (3)

- 接著撰寫對應的 ViewSwitcherController

```
public class ViewSwitcherController : Controller
{
    public RedirectResult SwitchView(bool mobile, string returnUrl)
    {
        if (Request.Browser.IsMobileDevice == mobile)
            HttpContext.ClearOverriddenBrowser();
        else
            HttpContext.SetOverriddenBrowser(mobile ? BrowserOverride.Mobile : BrowserOverride.Desktop);

        return Redirect(returnUrl);
    }

}
```

- 記得在 _Layout.mobile.cshtml 加上

```
<body>
    @Html.Partial("_ViewSwitcher")
```

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- **Task Support for Asynchronous Controllers**
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

新版非同步程式撰寫支援

- 新增可支援非同步作業傳回值的 `Task` 類別
- 在需要非同步執行的方法中使用 `async` 關鍵字
- 在實際需要等待傳回值的方法前使用 `await` 關鍵字來等待結果回傳。

```
public async Task<ActionResult> MyCustomers()
{
   var myApi = new HttpClient();
   myApi.BaseAddress = new Uri("http://localhost:1168");
   Task<HttpResponseMessage> response = myApi.GetAsync("api/MyCusAPI/");
   IEnumerable<Customers> result = await
response.Result.Content.ReadAsAsync<IEnumerable<Customers>>();
   return View(result);
}
```

新版非同步程式撰寫支援

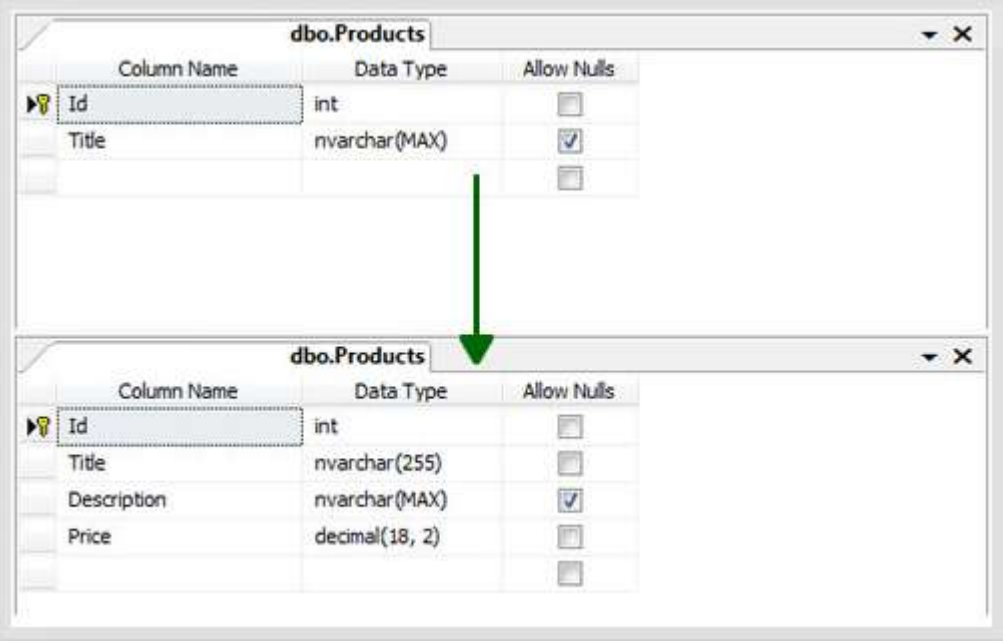
- 新增可支援非同步作業傳回值的 Task 類別
- 在需要非同步執行的方法中使用 async 關鍵字
- 在實際需要等待傳回值的方法前使用 await 關鍵字來等待結果回傳。

```
public async Task<ActionResult> MyCustomers()
{
    var myApi = new HttpClient();
    myApi.BaseAddress = new Uri("http://localhost:1168");
    Task<HttpResponseMessage> response = myApi.GetAsync("api/MyCusAPI/");
    IEnumerable<Customers> result = await
response.Result.Content.ReadAsAsync<IEnumerable<Customers>>();
    return View(result);
}
```

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- **Database Migrations**
- Empty Project Template
- Add Controller to any project folder
- Bundling and Minification

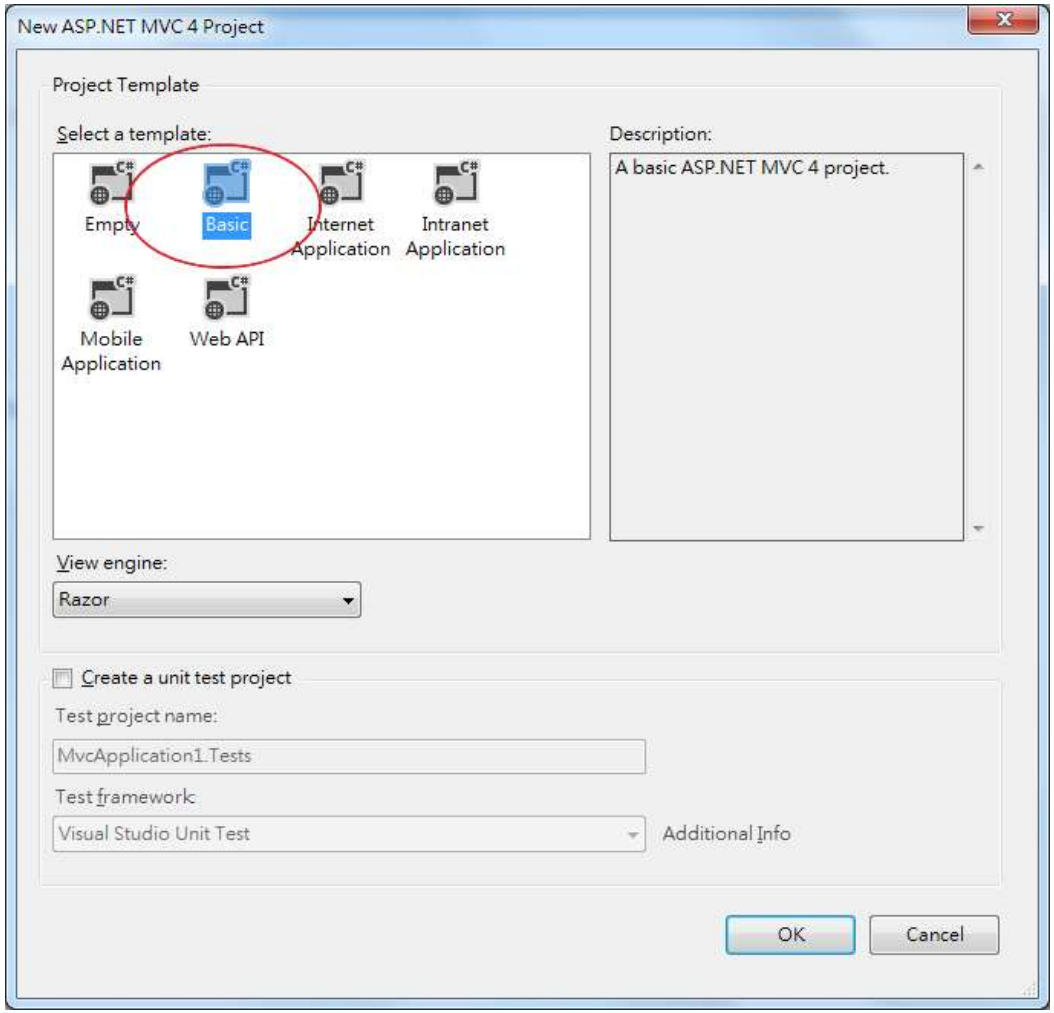
Database Migrations

- Code-First 功能再延伸
- 資料表Table不需要重新建立，動態加入新欄位
- 設定更簡單，更易於使用



如何建立Database Migrations的應用程式1

- 使用 Basic 範本這樣就不需要手動使用NuGet加入jQuery & jQuery UI & jQuery Validation 等套件。



如何建立Database Migrations的應用程式2

- 在 Models 資料夾下面建立 Products 類別 (資料表)

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace Mvc4DatabaseMigrationApp1.Models
7  {
8      public class Product
9      {
10         public int Id { get; set; }
11         public string Title { get; set; }
12     }
13 }
```

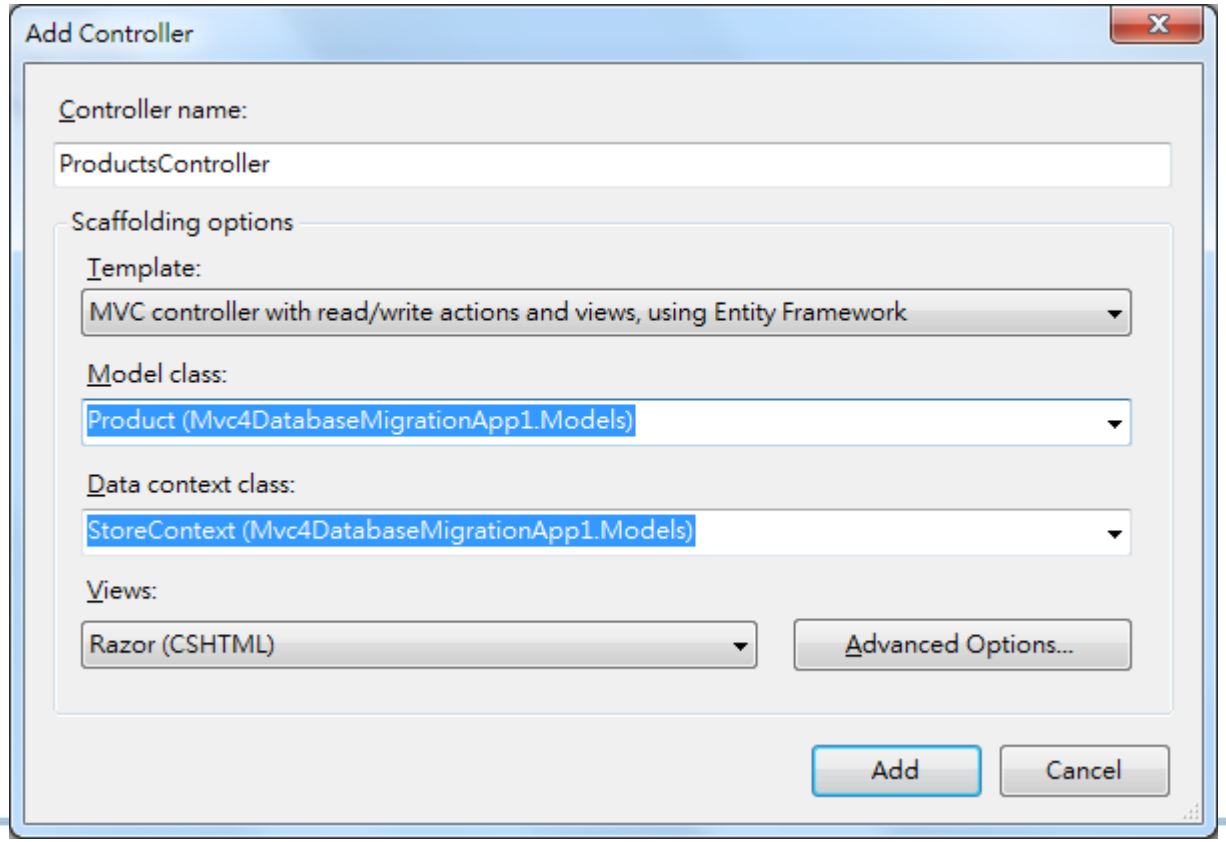
如何建立Database Migrations的應用程式3

- 建立 DbContext 類別

```
1  using System;
2  using System.Collections.Generic;
3  using System.Data.Entity;
4  using System.Linq;
5  using System.Web;
6
7  namespace Mvc4DatabaseMigrationApp1.Models
8  {
9      public class StoreContext: DbContext
10     {
11         public DbSet<Product> Production { get; set; }
12     }
13 }
```

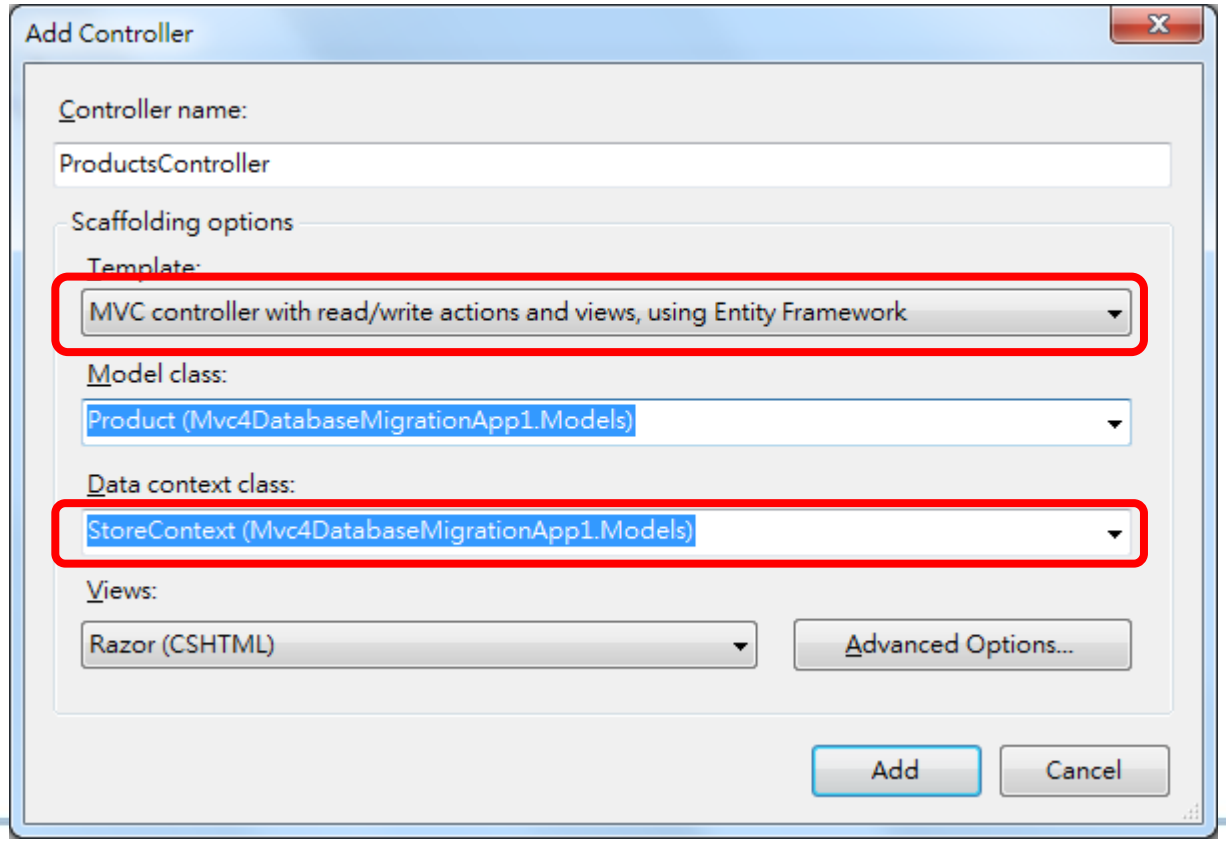
如何建立Database Migrations的應用程式4

- 編譯後、新增一個 ProductsController



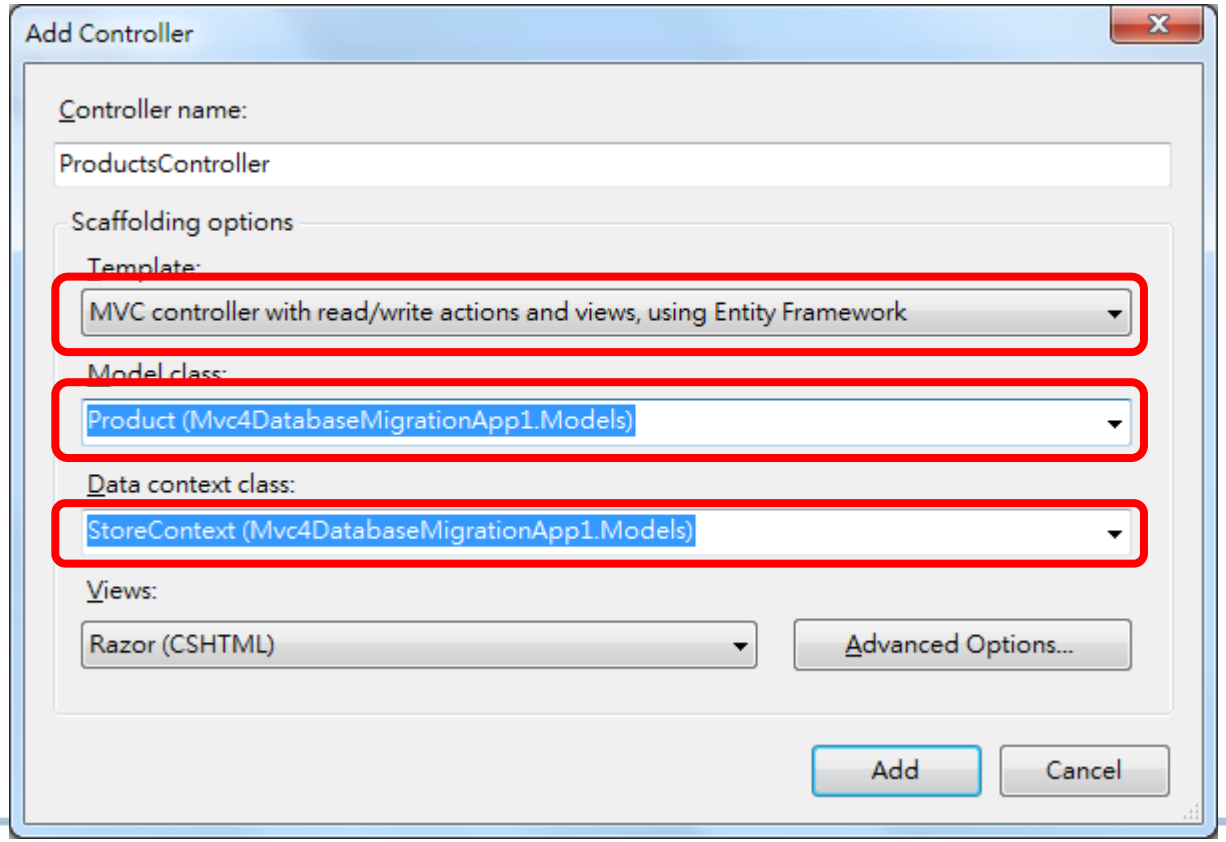
如何建立Database Migrations的應用程式4

- 編譯後、新增一個 ProductsController



如何建立Database Migrations的應用程式4

- 編譯後、新增一個 ProductsController



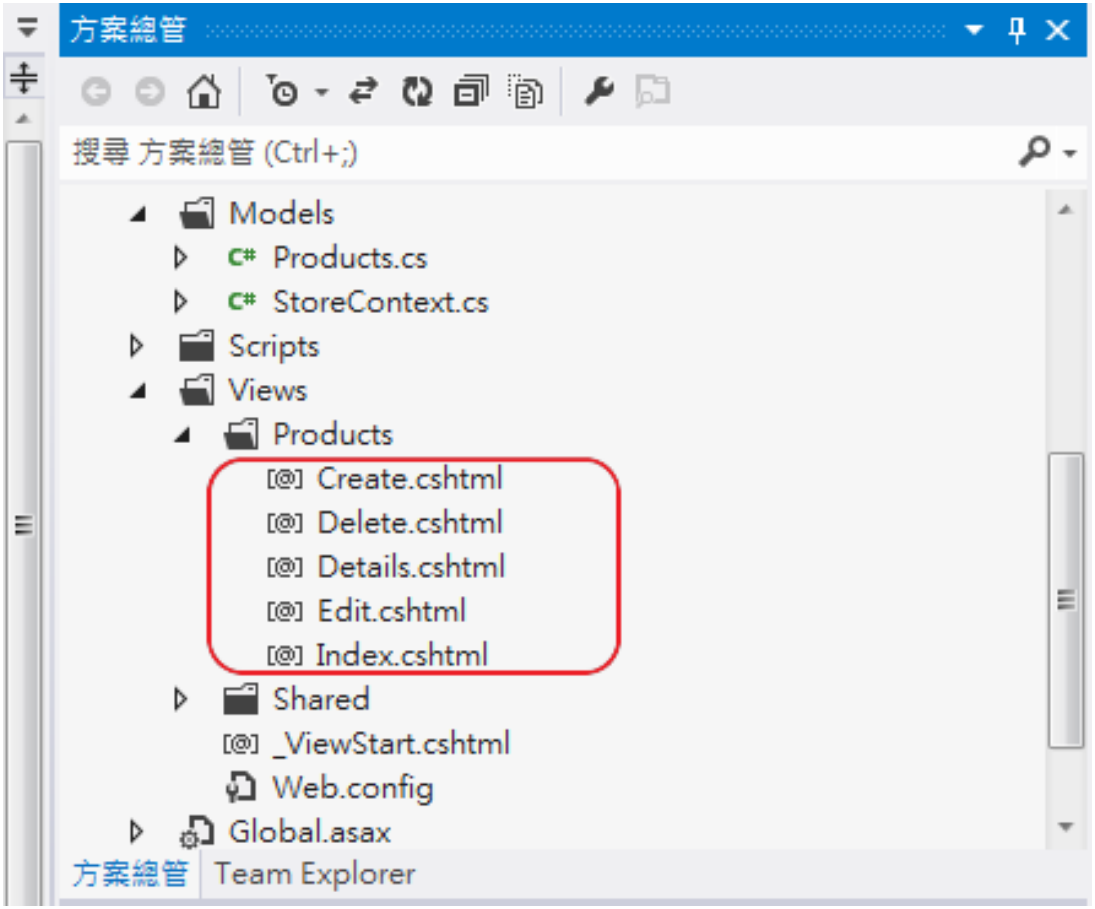
如何建立Database Migrations的應用程式5

- 預設VS2012 會幫你將資料儲存到

```
<connectionStrings>
  <add name="DefaultConnection" providerName="System.Data.SqlClient"
connectionString="Data Source=(LocalDb)\v11.0;Initial Catalog=aspnet-
Mvc4DatabaseMigrationApp2-20120729210048;Integrated
Security=SSPI;AttachDBFilename=|DataDirectory|\aspnet-
Mvc4DatabaseMigrationApp2-20120729210048.mdf" />
 </connectionStrings>
```

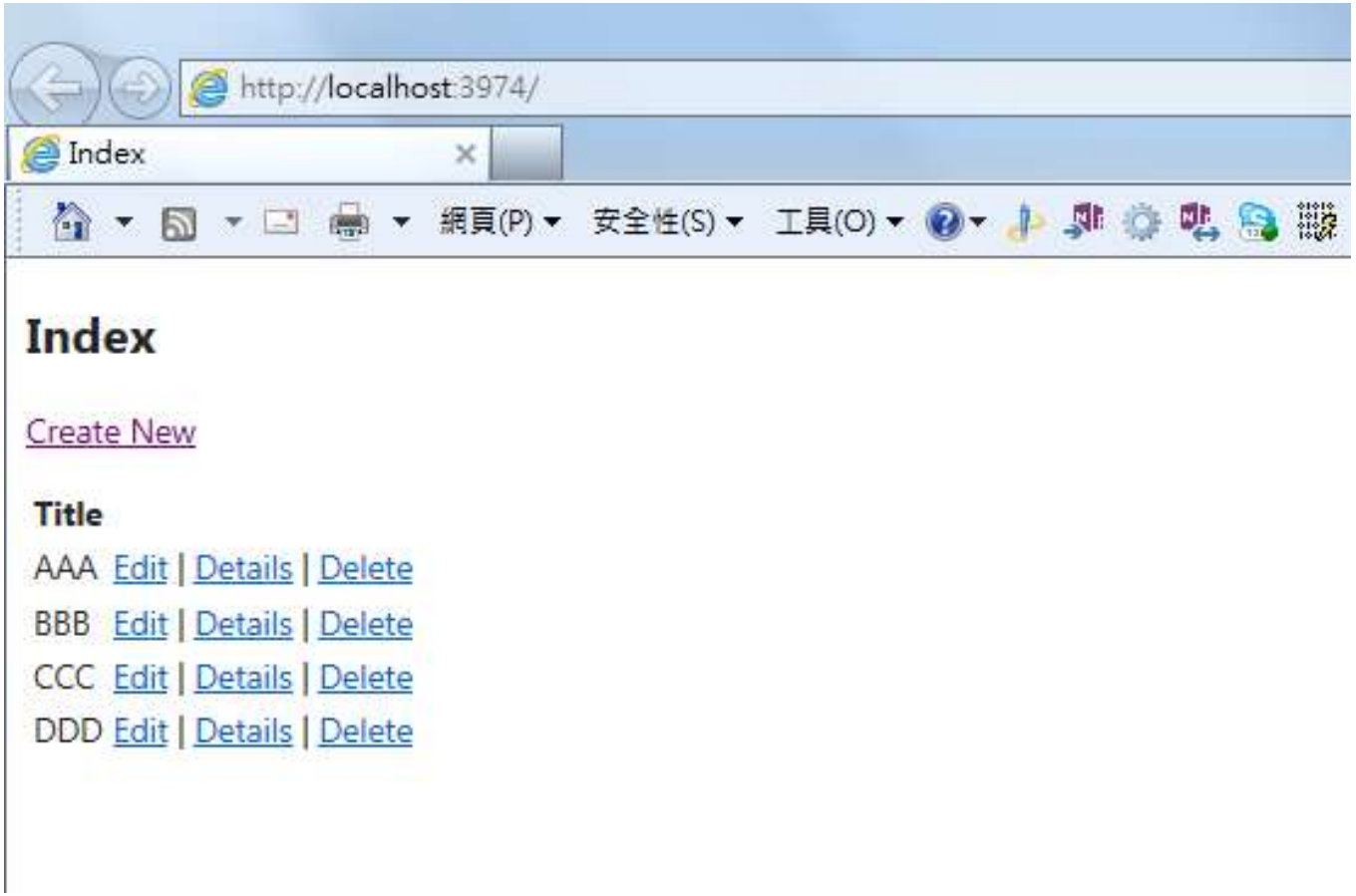
如何建立Database Migrations的應用程式6

- 程式立即可以執行
- Demo...



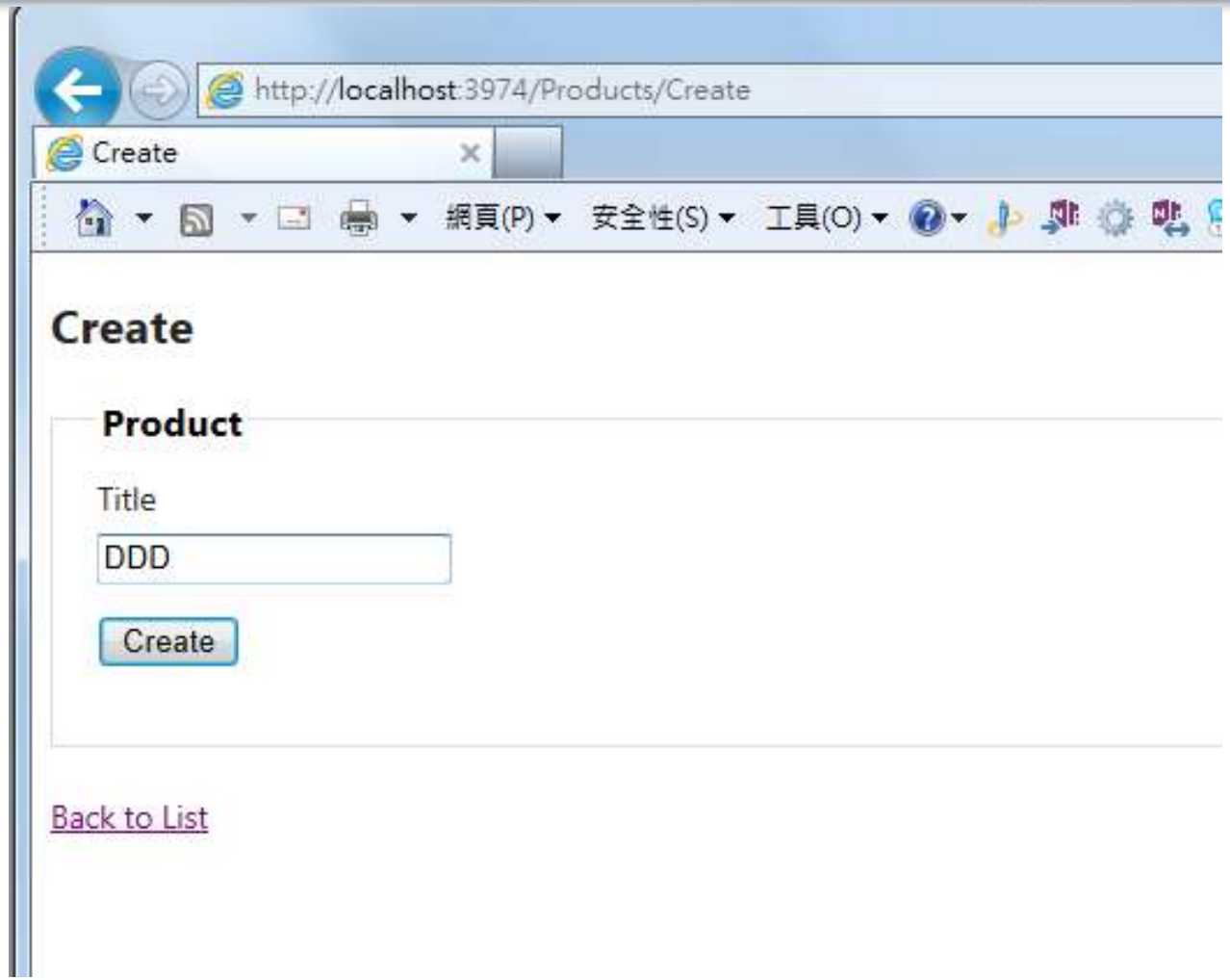
如何建立Database Migrations的應用程式6

- 程式立即可以執行
- Demo...



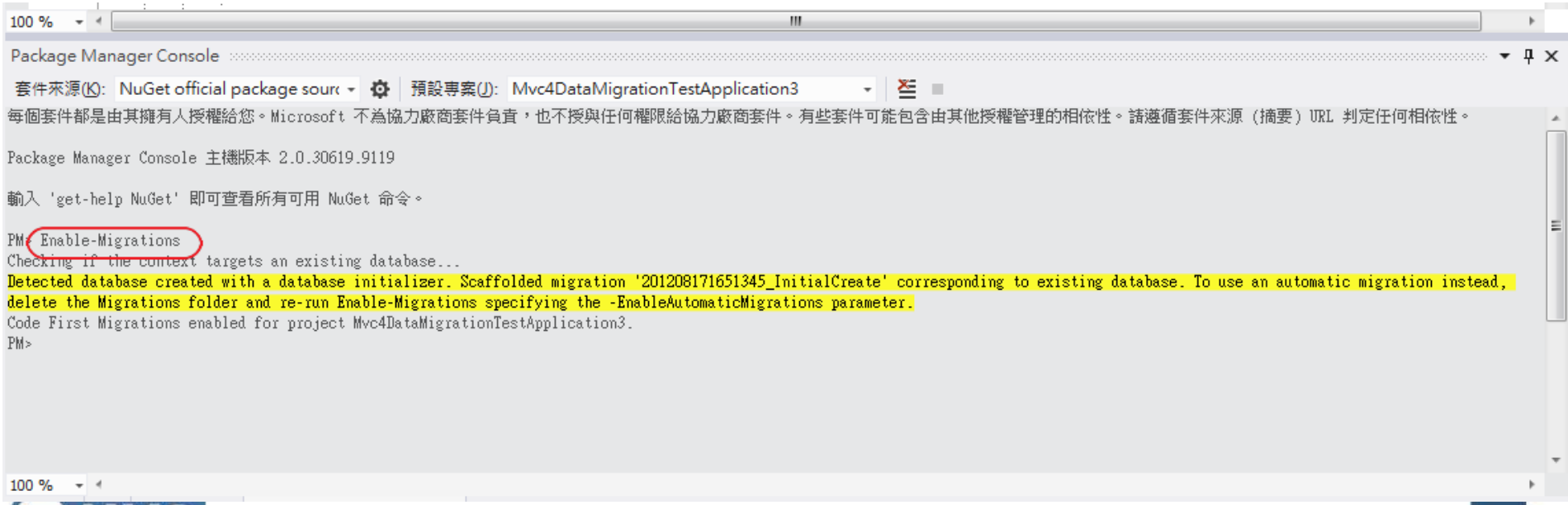
如何建立Database Migrations的應用程式6

- 程式立即可以執行
- Demo...



如何建立Database Migrations的應用程式7

- Data Migration 的主要功能為，在不重建、清除資料的情況下，動態刪減欄位。
- 透過Package Manager Console下Cmdlet指令來完成
- 首先須先開啟Migration功能：



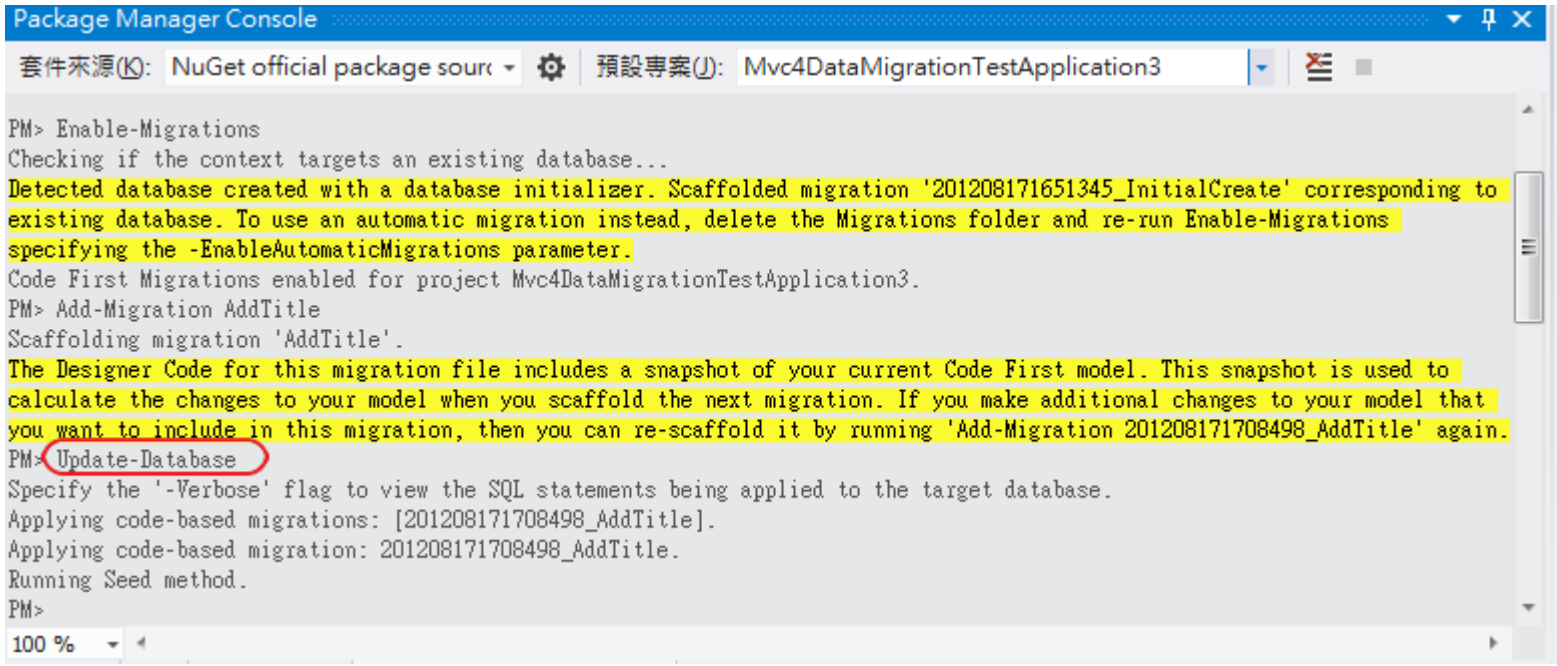
如何建立Database Migrations的應用程式8

- 加入實際的欄位



如何建立Database Migrations的應用程式9

- 最後一個CmdLet，真的將欄位加入到資料庫中。



如何建立Database Migrations的應用程式10

- Demo..

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- **Add Controller to any project folder**
- Bundling and Minification

- ASP.NET Web API
- Enhancements to Default Project Templates
- Mobile Project Template
- Display Modes
- jQuery Mobile, the View Switcher, and Browser Overriding
- Task Support for Asynchronous Controllers
- Database Migrations
- Empty Project Template
- Add Controller to any project folder
- **Bundling and Minification**

Bundling and Minification

- 在HTTP 中傳輸 JavaScript 或 CSS 的時候，它會幫您一除掉一些不必要 [字元] 與 [空白]
- 使用方式..

```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);

        BundleConfig.RegisterBundles(BundleTable.Bundles);
    }
}
```

Bundling and Minification

- 在HTTP 中傳輸 JavaScript 或 CSS 的時候，它會幫您一除掉一些不必要 [字元] 與 [空白]
- 使用方式..

```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();

        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);

        BundleConfig.RegisterBundles(BundleTable.Bundles);
    }
}
```

Bundling and Minification

- 在HTTP 中傳輸 J
  除掉一些不必要 [
- 使用方式..

```
public class MvcApplication : Sy
{
    protected void Application_S
    {
        AreaRegistration.Registe

        FilterConfig.RegisterGlo
        RouteConfig.RegisterRout

        BundleConfig.RegisterBun
    }
}
```

```
public class BundleConfig
{
    public static void RegisterBundles(BundleCollection bundles)
    {
        bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
                    "~/Scripts/jquery-1.*"));

        bundles.Add(new ScriptBundle("~/bundles/jqueryui").Include(
                    "~/Scripts/jquery-ui*"));

        bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
                    "~/Scripts/jquery.unobtrusive*",
                    "~/Scripts/jquery.validate*"));

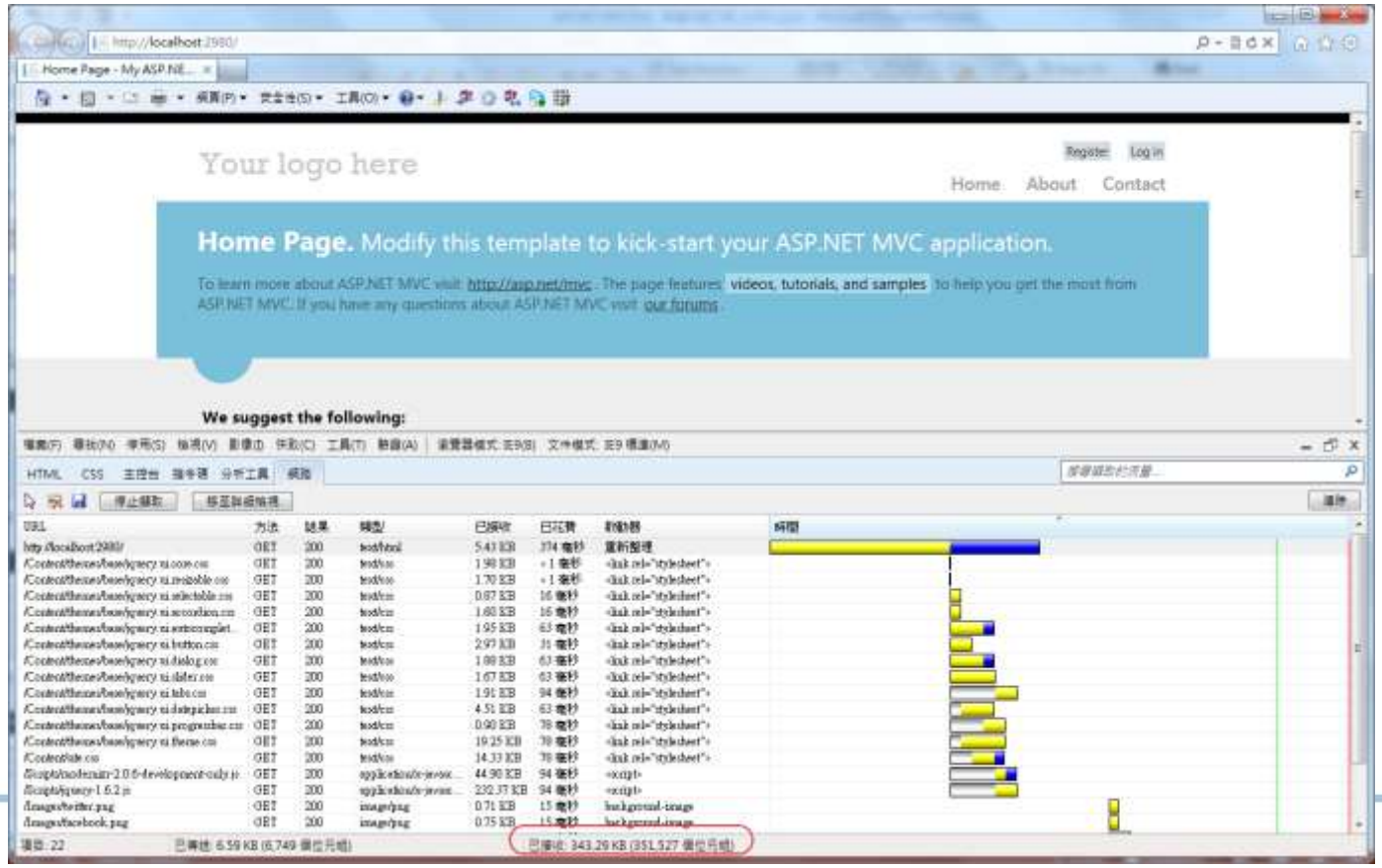
        bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
                    "~/Scripts/modernizr-*"));

        bundles.Add(new StyleBundle("~/Content/css").Include("~/Content/site.css

        bundles.Add(new StyleBundle("~/Content/themes/base/css").Include(
                    "~/Content/themes/base/jquery.ui.core.css",
                    "~/Content/themes/base/jquery.ui.resizable.css",
                    "~/Content/themes/base/jquery.ui.selectable.css",
                    "~/Content/themes/base/jquery.ui.accordion.css",
                    "~/Content/themes/base/jquery.ui.autocomplete.css",
                    "~/Content/themes/base/jquery.ui.button.css",
                    "~/Content/themes/base/jquery.ui.dialog.css",
                    "~/Content/themes/base/jquery.ui.slider.css",
                    "~/Content/themes/base/jquery.ui.tabs.css",
                    "~/Content/themes/base/jquery.ui.datepicker.css",
                    "~/Content/themes/base/jquery.ui.progressbar.css",
                    "~/Content/themes/base/jquery.ui.theme.css"));
    }
}
```

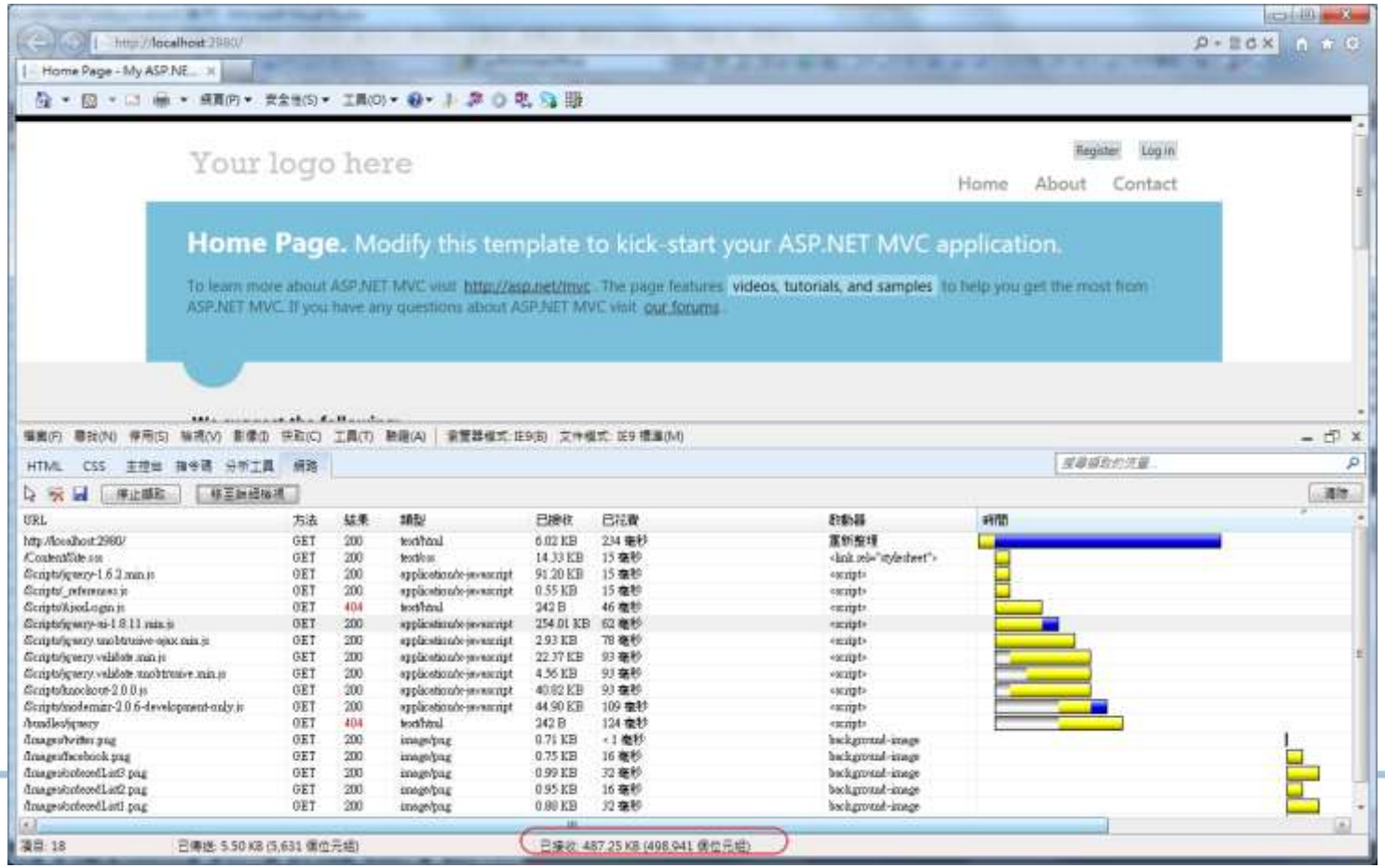
Bundling and Minification (2)

- 使用預設的BundleTable的壓縮功能



Bundling and Minification (3)

- 不使用BundleTable的壓縮功能



Bundling and Minification

- Demo…

聯絡資訊



- Blog: http://www.dotblogs.com.tw/gelis/
- MSN: wergelis@msn.com
- Facebook: http://fb.me/gelis.wu

Q&A

twMVC

- 官網：http://mvc.tw
- 臉書：http://fb.me/twmvc
- G+ ：http://mvc.tw/+