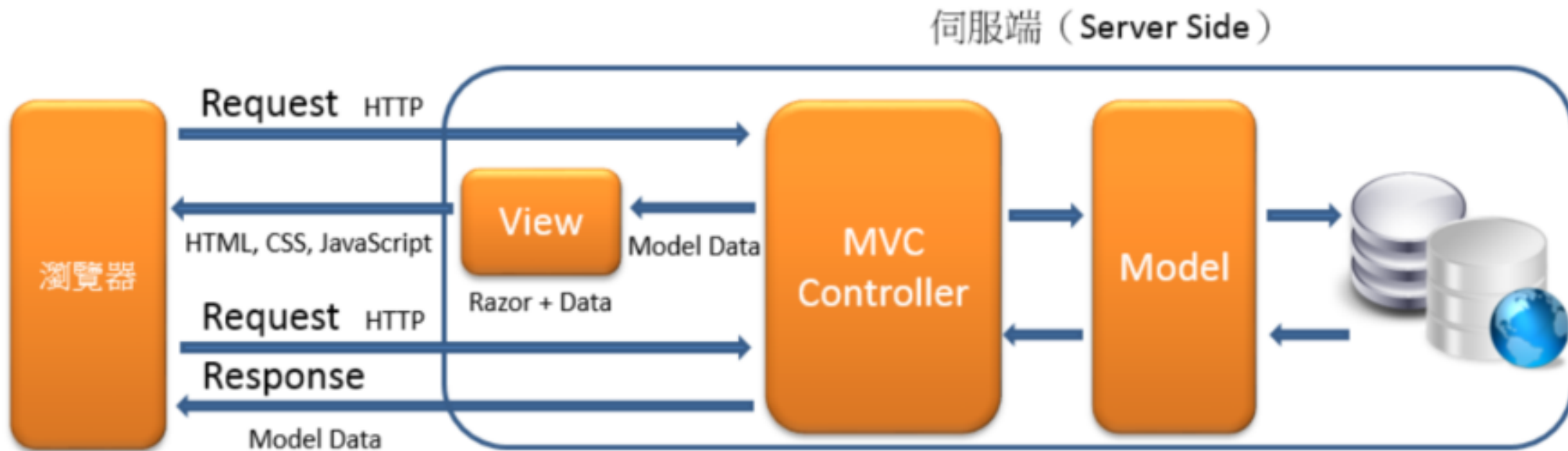


MVC5與Extension整合實作

Jacky

面對的技術選擇：MVC架構範圍



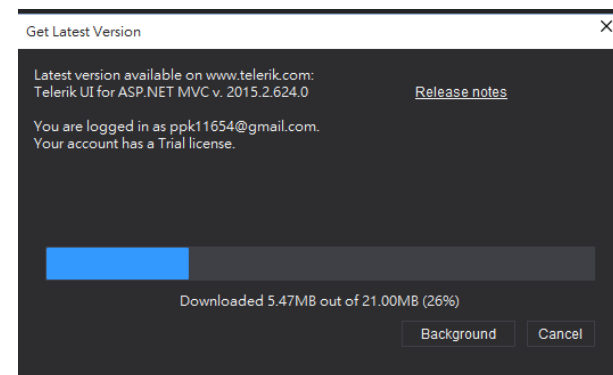
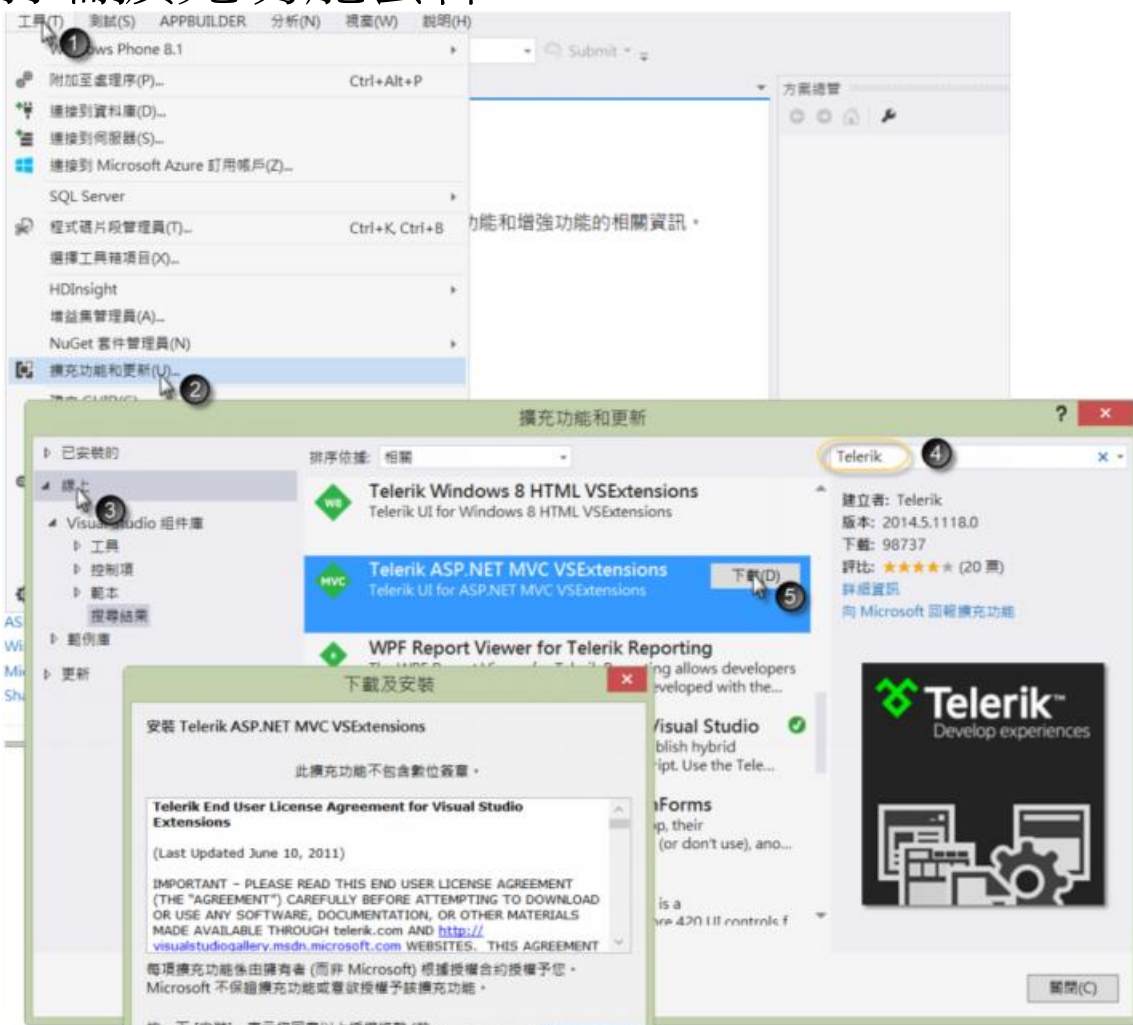
需求範例

- 在購物網站的首頁，我們想放一個影片，因為影片的內容有季節性，或是因為某項促銷活動而希望能在某一段特定期間內撥放，且又因為伺服器空間和網路流量的問題，因而希望這些影片是透過 YouTube 來播放。
- 為了達成這項需求，首先想到的可能是先要有一個類似的後台維護介面，讓購物網站的管理者可以新增、編輯、刪除來管理欲顯示在首頁的影片。



開發工具

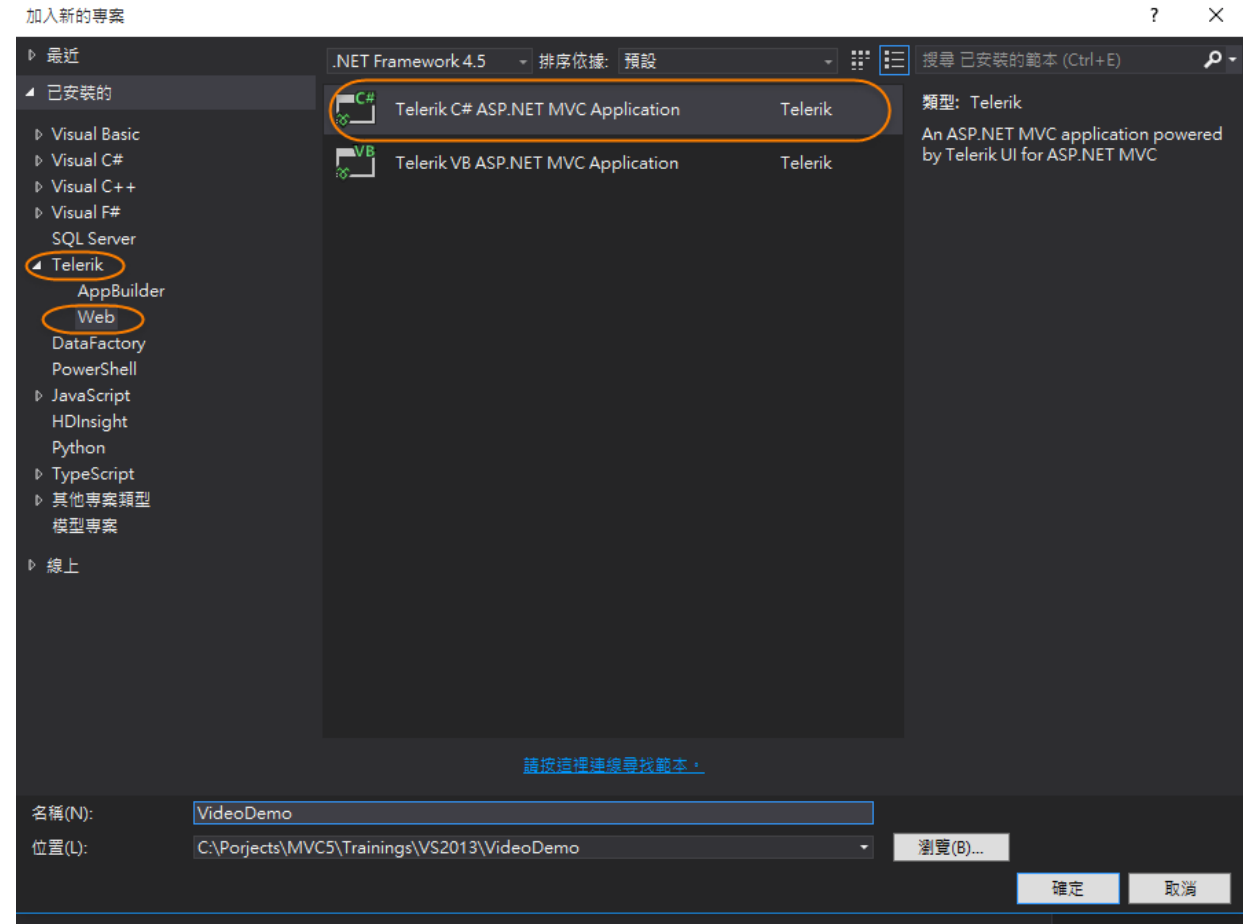
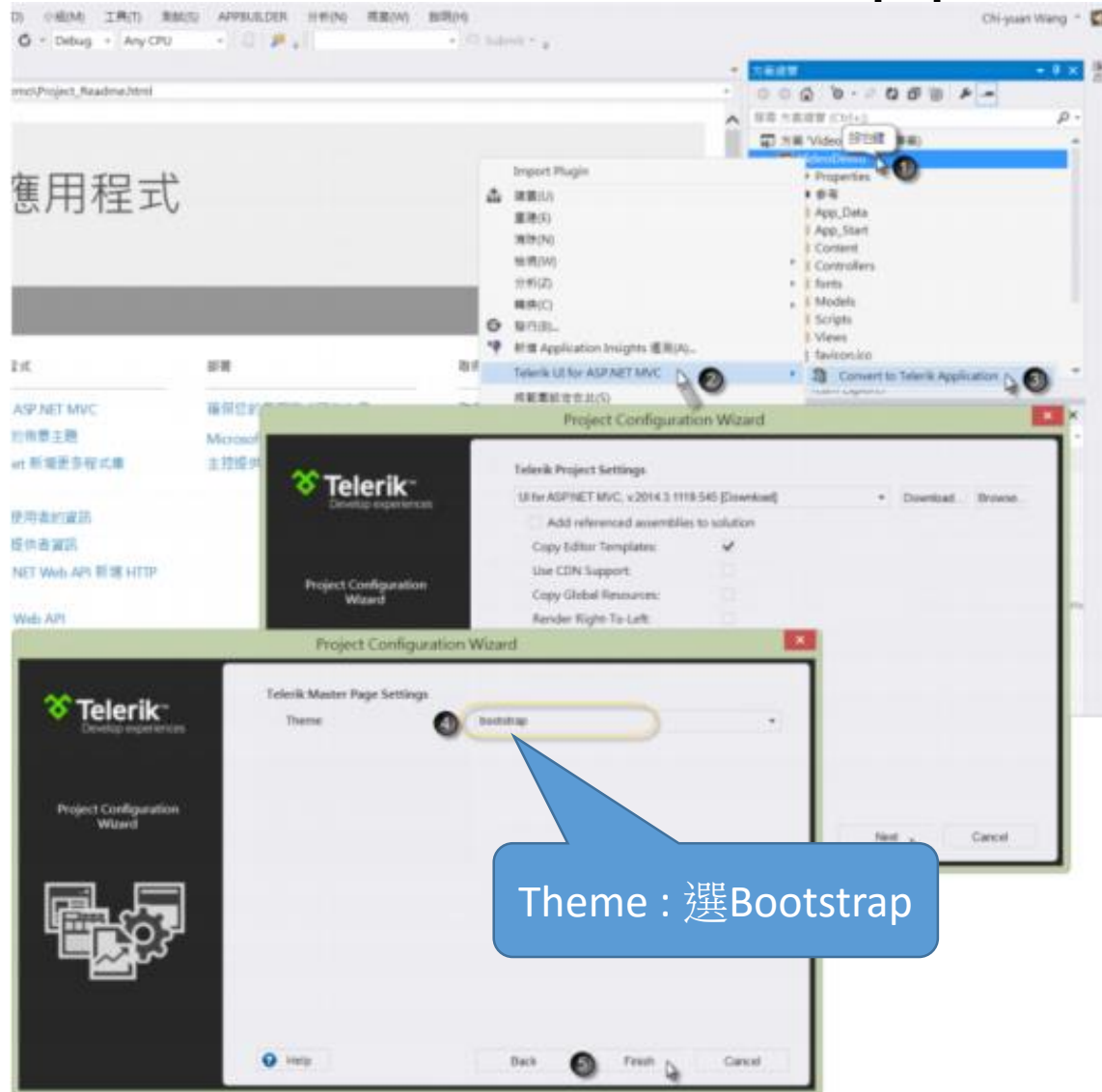
- 開發工具:VS 2013以上(免費版Visual Studio Community 2013+中文語言套件)
- 前端擴充功能套件:Telerik UI for ASP.NET MVC (*是付費套件)



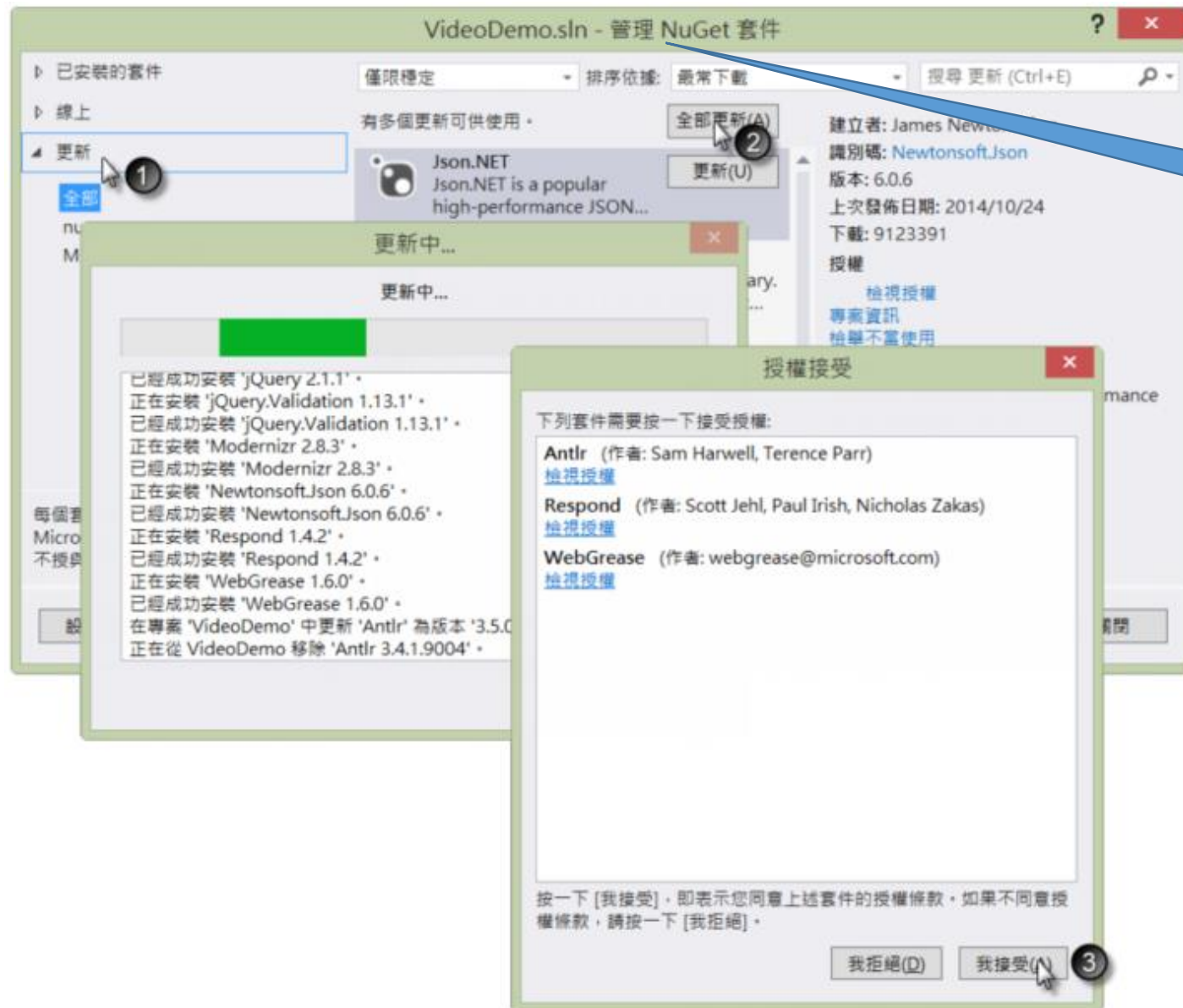
建立 ASP.NET MVC 專案 : VideoDemo



Telerik UI for ASP.NET MVC 的相關套件加入- Convert to Telerik Application

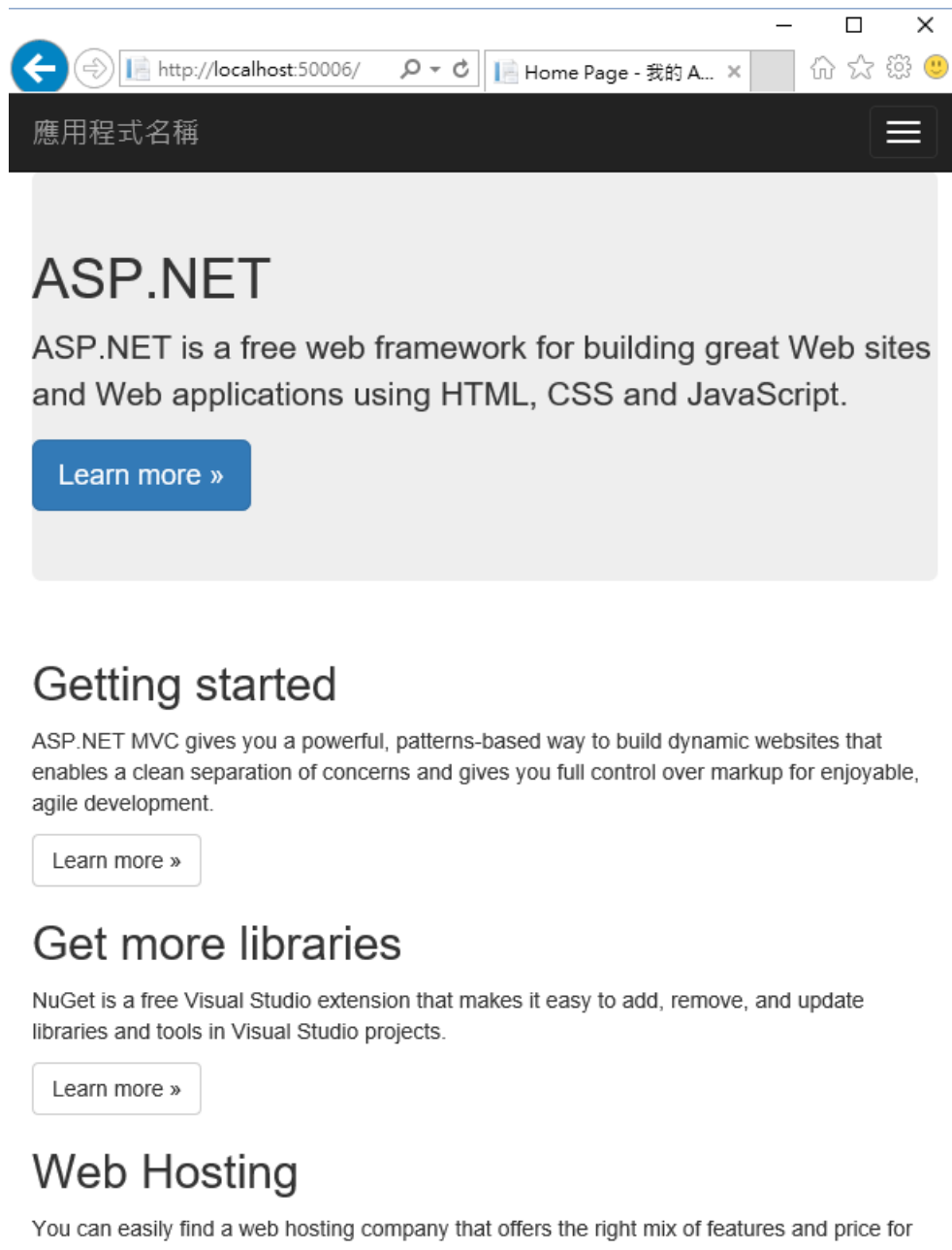


NuGet 套件管理員-管理方案套件管理



- [已安裝]→[全部]:解除不需要的套件
- [更新]→[全部]:更新用到所有的套件

F5先執行看看



資料模型-系統分析YouTube影片

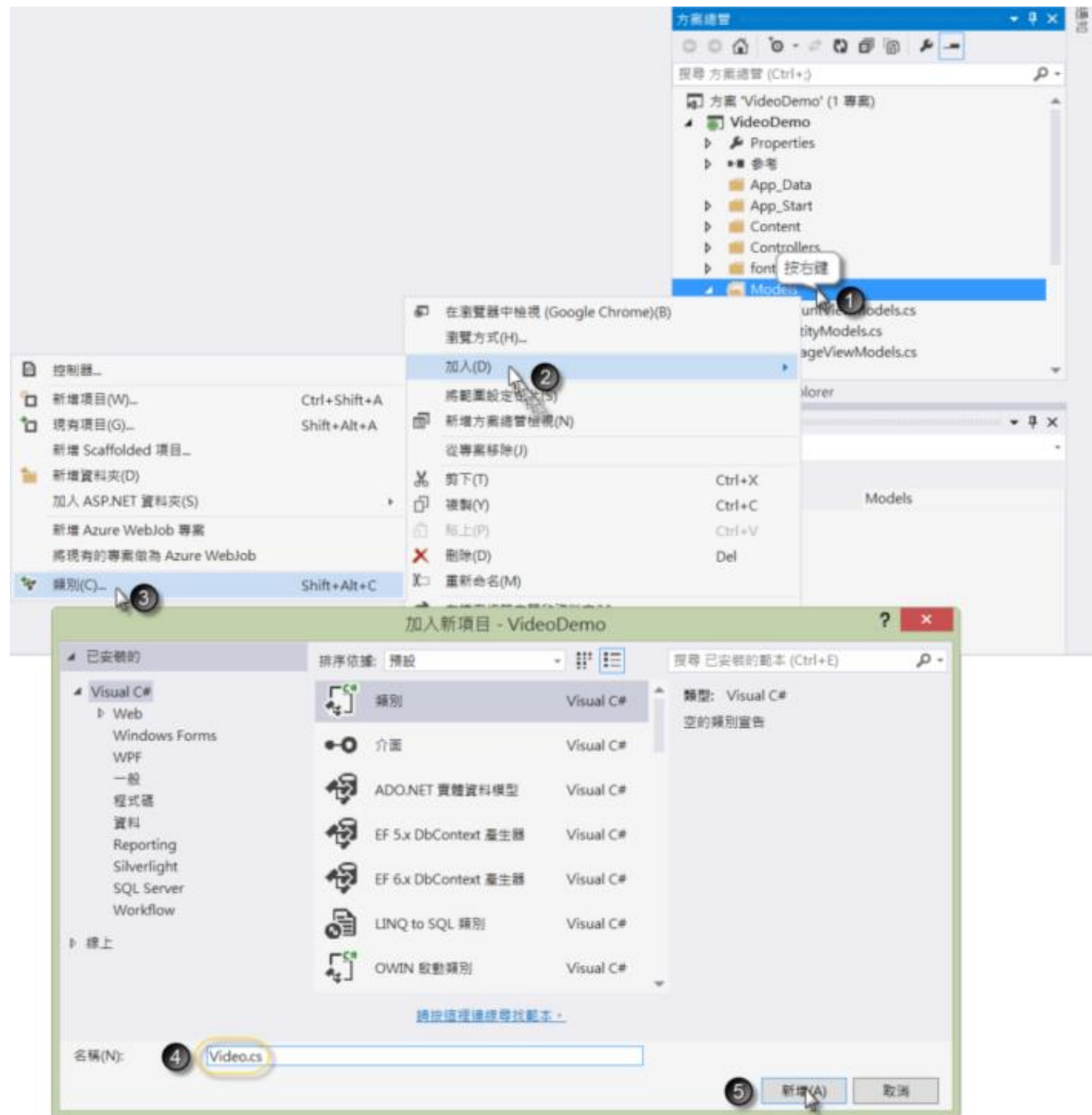


系統分析一下：

- 欲將 YouTube 嵌入網頁的程式碼，知道了由唯一的 11 個字元所組成的字串，可以代表一部 YouTube 影片 (採購便利貼：<https://youtu.be/oZ76aTT5cy8>)，再加上影片要在特定的期間內播放，所以[開始日期]與[結束日期]是資料模型所需要的屬性，為了方便管理，再給一個影片[標題]
- 習慣上資料模型都會給一個主鍵值 (Primary Key)，而[影片代碼]正好適合用來當作主鍵值 (預設主鍵值的屬性名稱為 Id)

實作-資料模型

- Models加入一個類別



資料模型程式碼-1

- 定義屬性
- 屬性上頭加上修飾用語（ **Data Annotations** ）
- 介面是用哪一個樣版（ **Template** ）

```
using System.ComponentModel.DataAnnotations;
```

```
namespace VideoDemo.Models
{
    public class Video
    {
        [Display(Name = "影片代碼")]
        [Required(ErrorMessage = "{0}不可空白")]
        [StringLength(11, MinimumLength = 11, ErrorMessage = "{0}必須是{1}個字元")]
        [UIHint("Video")]
        public string Id { get; set; }

        [Display(Name = "影片標題")]
        [Required(ErrorMessage = "{0}不可空白")]
        [MaxLength(20)]
        public string Title { get; set; }

        [UIHint("Date")]
        [Display(Name = "開始日期")]
        public DateTime StartDate { get; set; }

        [UIHint("Date")]
        [Display(Name = "結束日期")]
        public DateTime EndDate { get; set; }
    }
}
```

資料模型程式碼-2

屬性上頭加上修飾用語 (Data Annotations)

The screenshot displays a web application for managing videos. The main interface includes a table with columns for video ID, title, start date, and end date. A modal window titled "Add new record" is open, showing input fields for these attributes. A red arrow points from the "影片代碼" (Video ID) field in the modal to a code snippet. The code snippet defines a `Video` class with data annotations for each property.

Modal Form Fields:

- 影片代碼: JmKTfcRz0FY
- 影片標題: 凡人
- 開始日期: 2013/1/1
- 結束日期: 2013/1/1

Code Snippet:

```
public class Video
{
    [Display(Name = "影片代碼")]
    [Required(ErrorMessage = "{0}不可空白")]
    [StringLength(11, MinimumLength = 11, ErrorMessage = "{0}必須是11個字元")]
    [UIHint("Video")]
    public string Id { get; set; }

    [Display(Name = "影片標題")]
    [Required(ErrorMessage = "{0}不可空白")]
    [MaxLength(20)]
    public string Title { get; set; }

    [UIHint("Date")]
    [Display(Name = "開始日期")]
    public DateTime StartDate { get; set; }

    [UIHint("Date")]
    [Display(Name = "結束日期")]
    public DateTime EndDate { get; set; }
}
```

實作-DbContext 類別

- Models下加入一個類別



DbContext程式碼

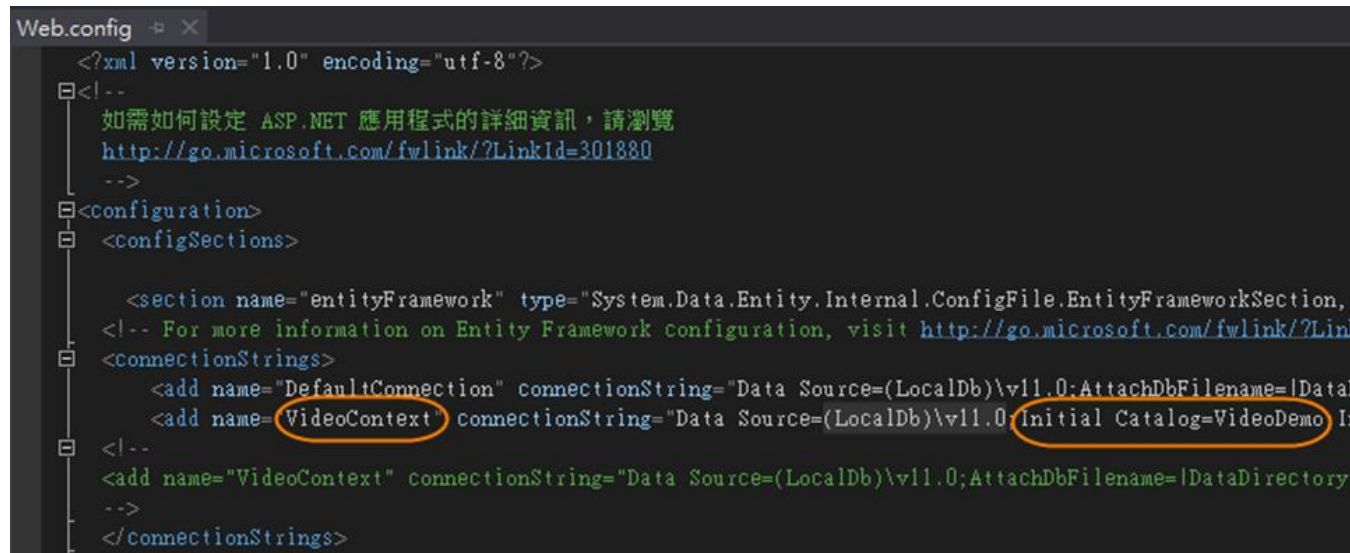
- 首先， VideoContext 繼承 DbContext 基底類別，可以想像這是要對應成一個資料庫。
- Videos 以 DbSet<> 型別為屬性可以視為資料表。
- 而 DbSet<> 內所包含的 Video 代表定義資料表內欄位名稱和型別的類別。

```
using System.Data.Entity;

namespace VideoDemo.Models
{
    public class VideoContext : DbContext
    {
        public DbSet<Video> Videos { get; set; }
    }
}
```


資料庫

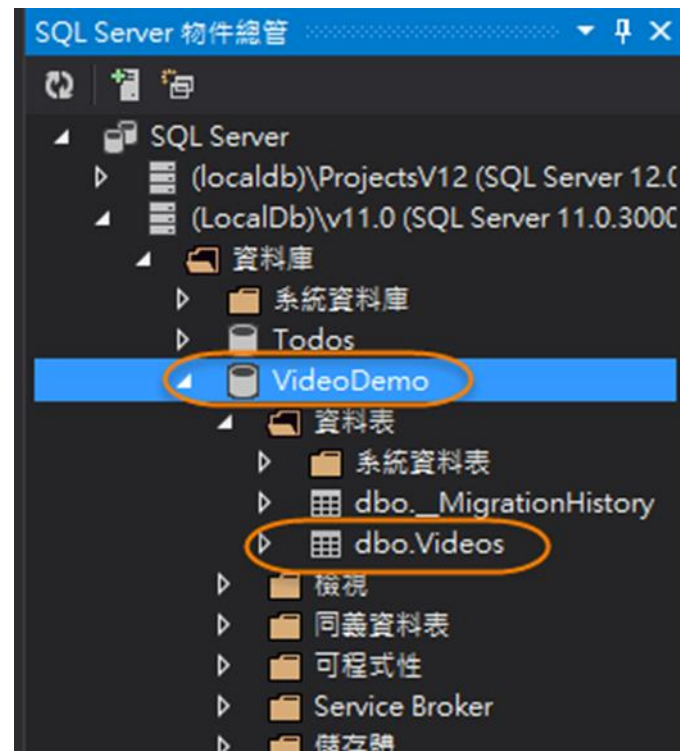
- 商業應用程式免不了需要一個資料庫用來儲存資料，既然先前實作的 DbContext 類別代表了資料庫與資料模型之間的對應（Object Relational Mapping : ORM）接下來就是指定一個資料庫建立該些對應關係（也就是建立資料表）。
- 而資料庫的指定，不外乎就就要讓應用程式知道如何去連它，一般會把相關資訊的連線字串寫在 Web.config 檔案中



Web.config

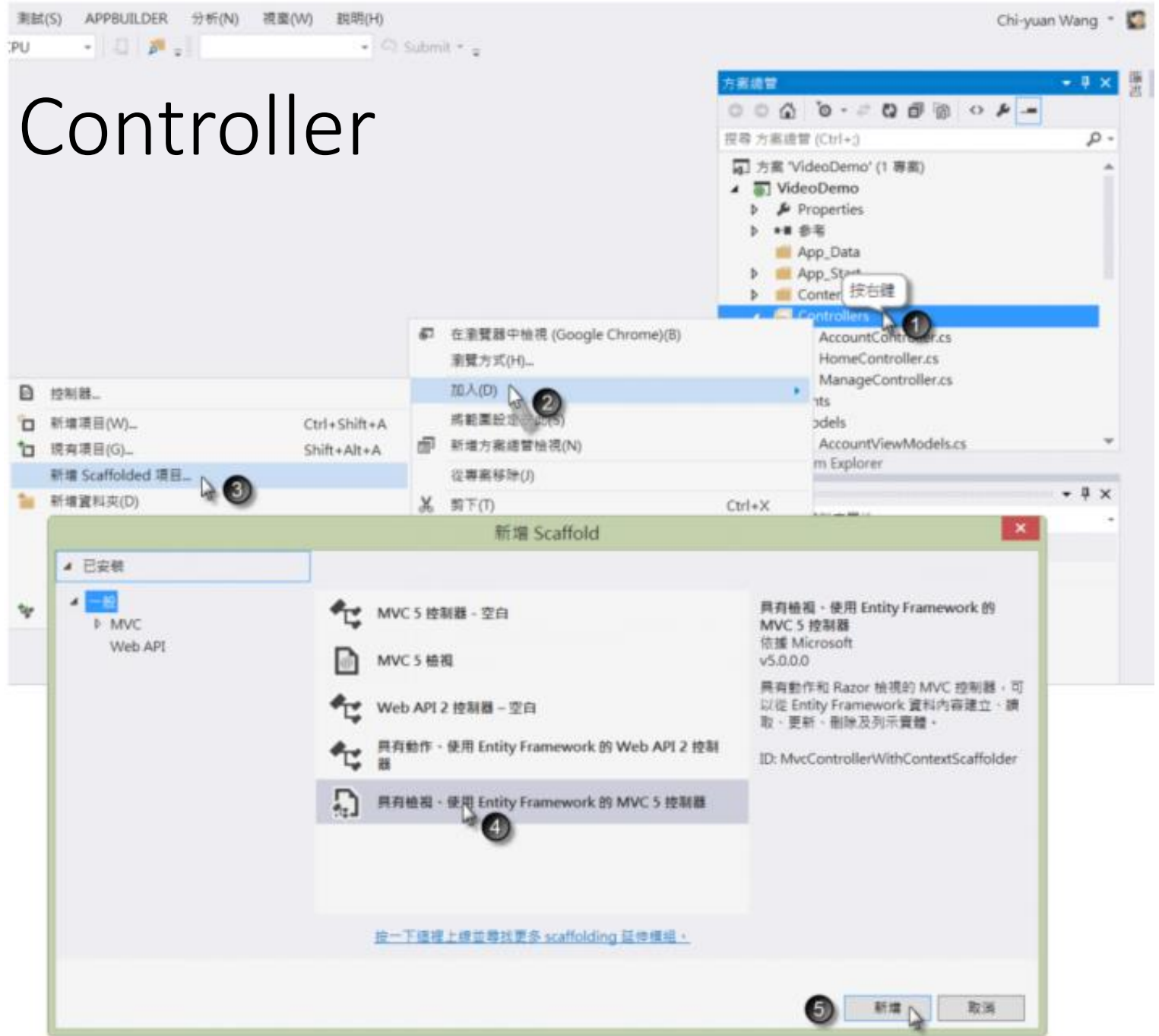
```
<?xml version="1.0" encoding="utf-8"?>
<!--
如需如何設定 ASP.NET 應用程式的詳細資訊，請瀏覽
http://go.microsoft.com/fwlink/?LinkId=301880
-->
<configuration>
  <configSections>

    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?Lin
  </section>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|Database1.sdf" providerName="Microsoft.Data.SqlClient" />
    <add name="VideoContext" connectionString="Data Source=(LocalDb)\v11.0;Initial Catalog=VideoDemo;Integrated Security=True" providerName="Microsoft.Data.SqlClient" />
  </connectionStrings>
</configuration>
```



ASP.NET MVC Controller

- ASP.NET MVC 最核心的部分：資料記錄的新增、讀取、更新、刪除（CRUD）操作
- 使用 scaffolding 的方式，利用工具來自動產生所需的程式碼
- 加入【具有檢視、使用 Entity Framework 的 MVC 5 控制器】

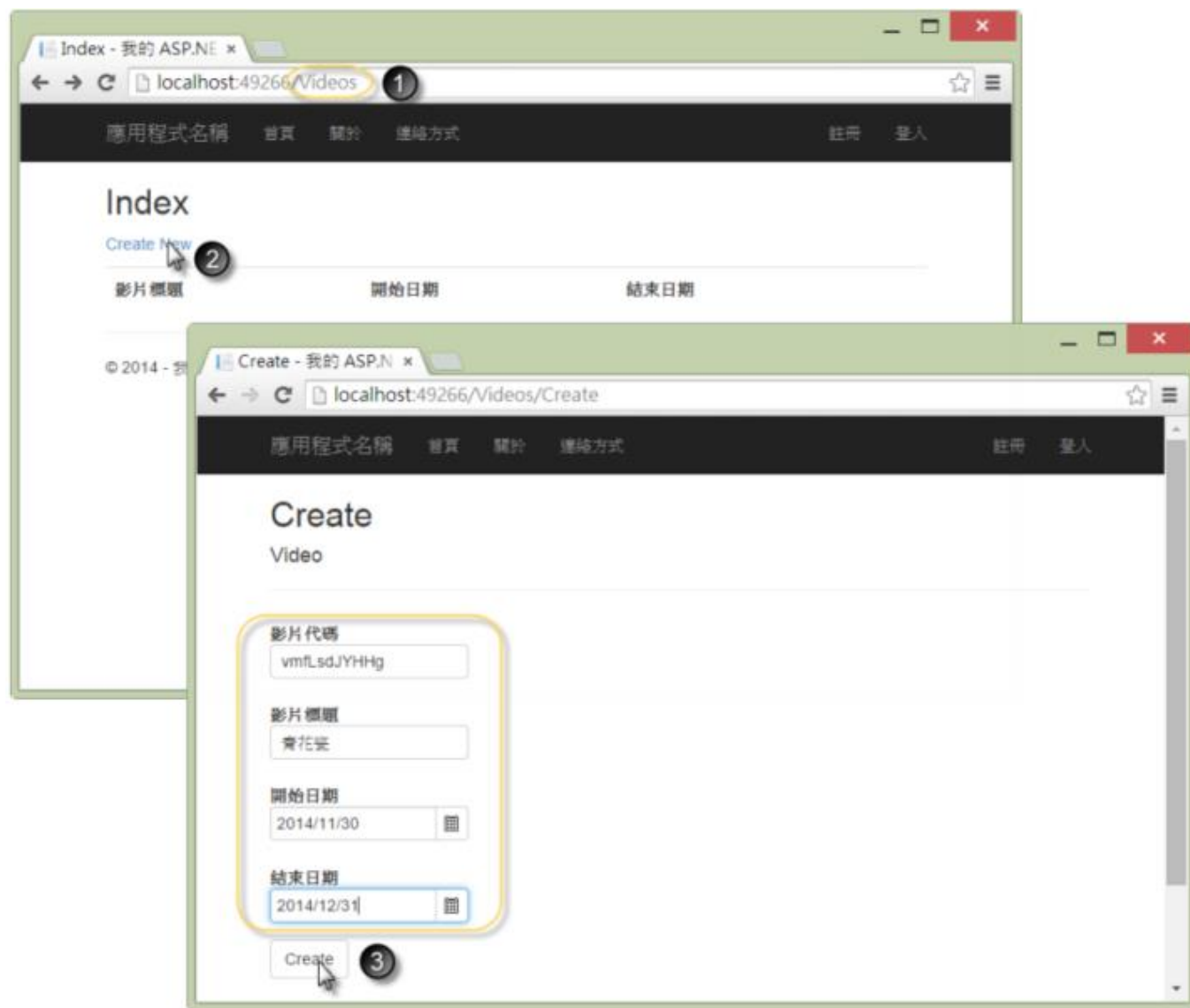


加入控制器

- 自動產生所有需要的控制器方法（**Controller Method**）以及檢視（**View**）的程式碼

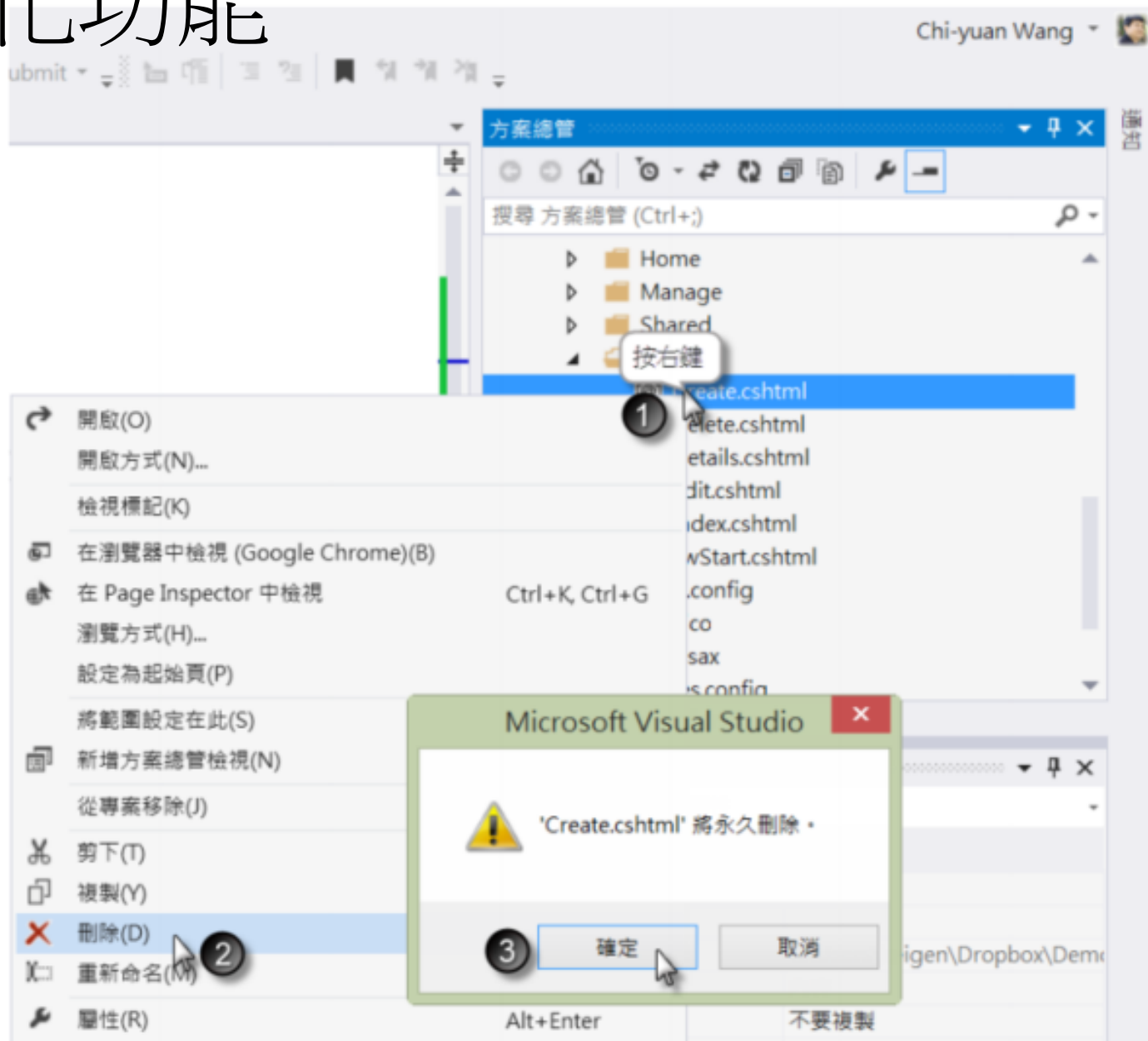


F5再執行看看:增刪修基本功能



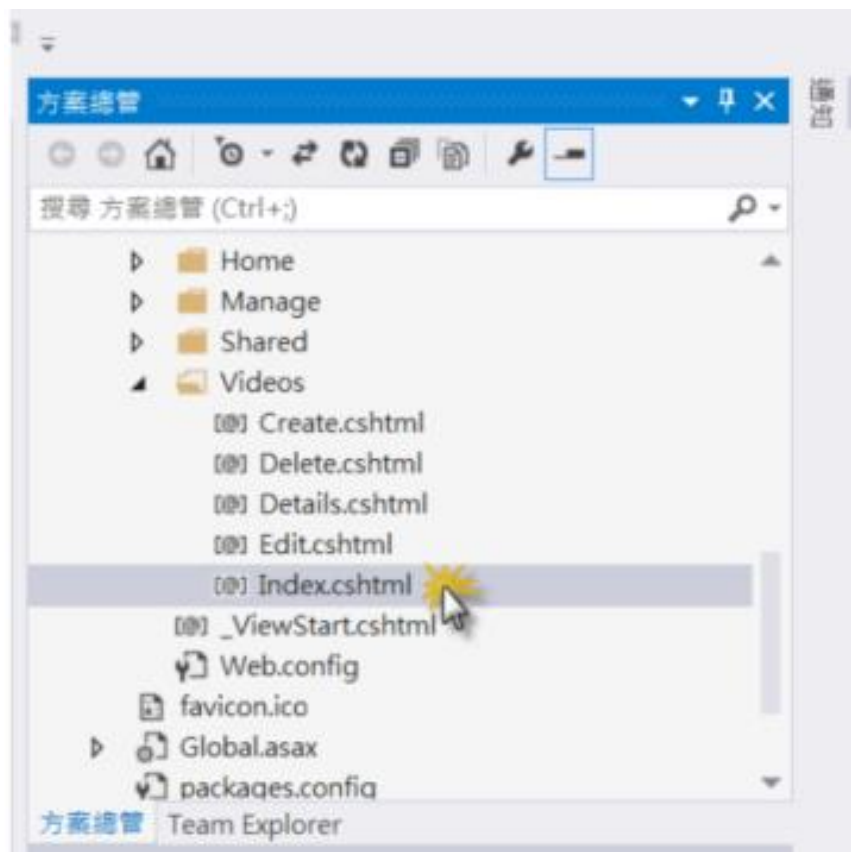
View改頭換面強化功能

- 使用 Grid 元件，只需要一個 View（本例是 Views\Videos\Index.cshtml）
- 其餘刪除



改造Index.cshtml

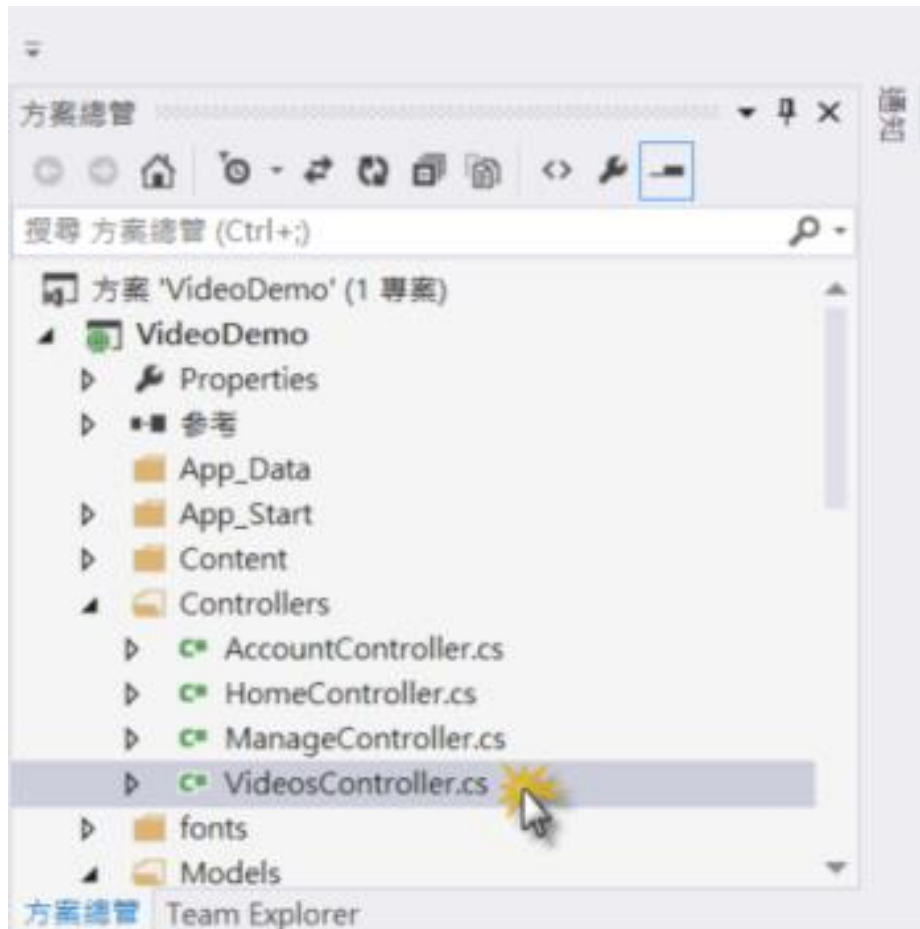
- Code註解說明



```
@model IEnumerable< VideoDemo. Models. Video>
@{
    ViewBag.Title = " Index" ;
}
@( Html. Kendo( ) . Grid( Model)
    . Name( " Videos" )
    . Columns( c =>
        {
            c. Command( p => p. Edit( ) . Text( "編輯" ) ) ;
            c. Bound( p => p. Id ) . Width( 310 ) ;
            c. Bound( p => p. Title ) ;
            c. Bound( p => p. StartDate ) ;
            c. Bound( p => p. EndDate ) ;
            c. Command( p => p. Destroy( ) . Text( "刪除" ) ) ;
        })
    . Toolbar( toolbar => toolbar. Create( ) . Text( "新增" ) )
    . Editable( e => e. Mode( GridEditMode. PopUp ) )
    . DataSource( d => d
        . Server( )
        . Model( m =>
            {
                m. Id( " Id" ) ;
                m. Field( f => f. StartDate ) . DefaultValue( DateTime. Today ) ;
                m. Field( f => f. EndDate ) . DefaultValue( DateTime. Today. AddDays( 15 ) ) ;
            })
        . Create( " Create" , " Videos" )
        . Update( " Edit" , " Videos" )
        . Destroy( " Delete" , " Videos" )
    )
)@
section scripts {
    @Scripts. Render( " ~/bundles/j queryval" )
}
```

改造對應的控制器方法 (Controller Method)

- Code 註解說明



```
using System.Data.Entity;  
using System.Linq;  
using System.Web.Mvc;  
using VideoDemo.Models;  
using Kendo.Mvc.Extensions;
```

```
namespace VideoDemo.Controllers
```

```
{
```

```
    public class VideosController : Controller
```

```
    {
```

```
        private VideoContext db = new VideoContext();
```

```
        public ActionResult Index()
```

```
        {
```

```
            return View(db.Videos.ToList());
```

```
        }
```

```
        [HttpPost]
```

```
        public ActionResult Create([Bind(Include = "Id,Title,StartDate,EndDate")] Video video)
```

```
        {
```

```
            if (ModelState.IsValid)
```

```
            {
```

```
                db.Videos.Add(video);
```

```
                db.SaveChanges();
```

```
                return RedirectToAction("Index", this.GridRouteValues());
```

```
            }
```

```
            return View("Index", db.Videos.ToList());
```

```
        }
```

```
        [HttpPost]
```

```
        public ActionResult Edit([Bind(Include = "Id,Title,StartDate,EndDate")] Video video)
```

```
        {
```

```
            if (ModelState.IsValid)
```

```
            {
```

```
                db.Entry(video).State = EntityState.Modified;
```

```
                db.SaveChanges();
```

```
                return RedirectToAction("Index", this.GridRouteValues());
```

```
            }
```

```
            return View("Index", db.Videos.ToList());
```

```
        }
```

```
        [HttpPost]
```

```
        public ActionResult Delete(string id)
```


樣版:Display Template 和 Editor Template

```
public class Video
{
    [Display(Name = "影片代碼")]
    [Required(ErrorMessage = "{0}不可空白")]
    [StringLength(11, MinimumLength = 11, ErrorMessage = "{0}必須是{1}個字元")]
    [UIHint("Video")]
    public string Id { get; set; }

    [Display(Name = "影片標題")]
    [Required(ErrorMessage = "{0}不可空白")]
    [MaxLength(20)]
    public string Title { get; set; }

    [UIHint("Date")]
    [Display(Name = "開始日期")]
    public DateTime StartDate { get; set; }

    [UIHint("Date")]
    [Display(Name = "結束日期")]
    public DateTime EndDate { get; set; }
}
```

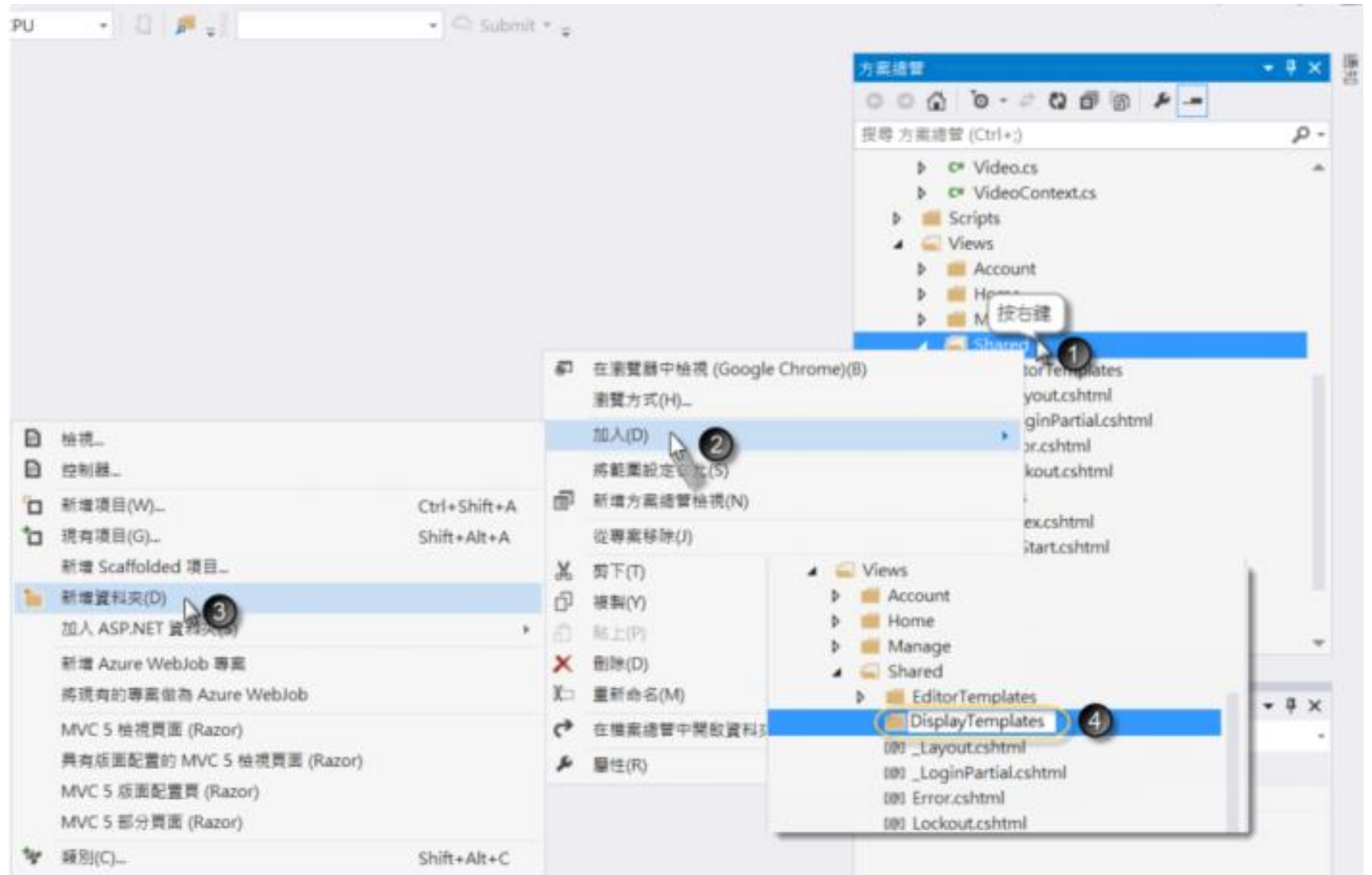
方案總管

搜尋 方案總管 (Ctrl+;)

- [@] Contact.cshtml
- [@] Index.cshtml
- Shared
- DisplayTemplates
 - [@] Date.cshtml
 - [@] Video.cshtml
- EditorTemplates
 - [@] Currency.cshtml
 - [@] Date.cshtml
 - [@] DateTime.cshtml
 - [@] Email.cshtml

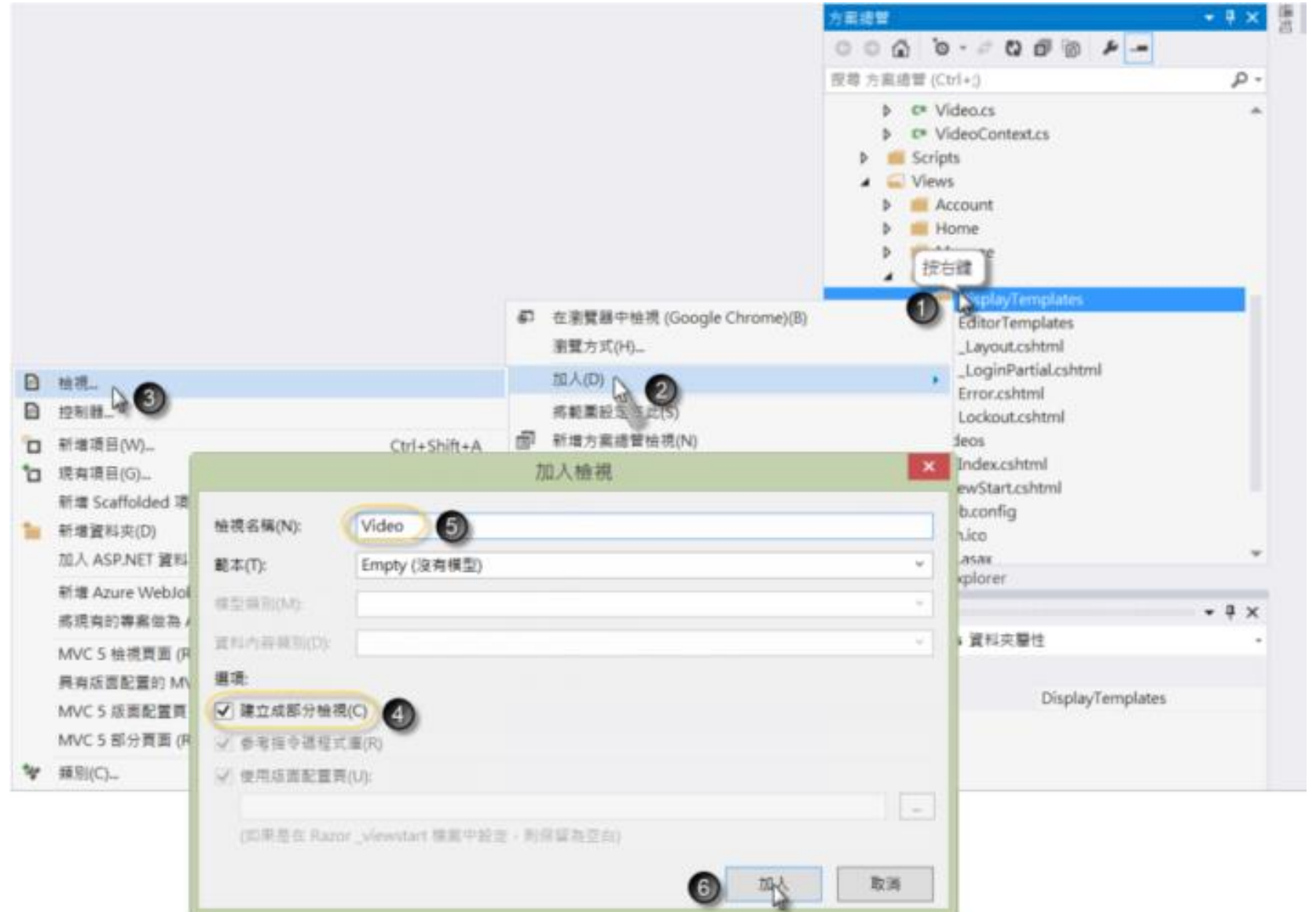
Display Template -1

- Views\Shared新增特定資料夾
DisplayTemplates



DisplayTemplates 新增一個部分檢視 -2

- Views\Shared\
DisplayTemplates
新增新增一個部
分檢視
Video



DisplayTemplates新增另一個部分檢視 -3

- Views\Shared\
DisplayTemplates
新增新增一個部
分檢視
- 同上一頁步驟檢
視名稱改為 **Date**

Views\Shared_Layout.cshtml 檔案內加入切換到影片 維護頁面的連結

@Html.ActionLink("連絡方式", "Contact", "Home") 如下程式碼所示：

```
< div class=" navbar- collapse collapse" >  
< ul class=" nav navbar- nav" >  
< li>@Html. ActionLink( "首頁", " Index", " Home" ) < /li>  
< li>@Html. ActionLink( "關於", " About", " Home" ) < /li>  
< li>@Html. ActionLink( "連絡方式", " Contact", " Home" ) < /li>  
< li>@Html. ActionLink( "影片維護", " Index", " Videos" ) < /li>  
< /ul>  
@Html. Partial( " _LoginPartial" )  
< /div>
```

控制影片 依所設定的日期期限首頁顯示:HomeController.cs

- \Controllers\HomeController.cs加上

.....

```
public ActionResult Index( )  
{  
var model =  
db. Videos  
. Where( e => e. StartDate <= DateTime. Today &&  
e. EndDate >= DateTime. Today)  
. OrderBy( e => Guid. NewGuid( ) ) . FirstOrDefault( ) ;  
return View(model) ;  
}.
```

影片 資料模型在首頁的 View 中顯示:Index.cshtml

- \Views\Home\Index.cshtml加上

```
.....  
< div class=" row" >  
    <div class="col-md-4">  
        @Html.DisplayFor(m => m.Id)  
    </div>  
.....
```

執行成果

←

→

http://localhost:50006/Videos

🔍

🔄

影片維護頁 - 我的 ASP.NET... x

🏠 ☆ ⚙️ 😊

應用程式名稱 首頁 關於 連絡方式 影片維護 註冊 登入

⊕ 新增

	影片代碼	影片標題	開始日期	結束日期	
<div>🕒 編輯</div>	<div><div>youshop 採購便利貼-您提供的多項服務</div><div><div>社群行銷</div><div>完善物流</div><div>安全金流</div><div>低手續費</div><div>客服處理</div></div></div>	youShop	2015/8/4	2015/12/31	<div>✕ 刪除</div>

© 2015 - 我的 ASP.NET 應用程式

結語

伺服器端 (Server Side)

