

RandomVariable / January 23, 2017 12:57AM

[Learn MVC Project in 7 days – Day 1 – Lab 2 – Demonstrating Views](https://www.codeproject.com/articles/866143/learn-mvc-step-by-step-in-days-day)

=====

[url=https://www.codeproject.com/articles/866143/learn-mvc-step-by-step-in-days-day]Learn MVC Project in 7 days – Day 1[/url]

=====

[url=https://www.codeproject.com/articles/866143/learn-mvc-project-in-days-day#Lab2%E2%80%93Demonstrating Views]Lab 2 – Demonstrating Views[/url]
Lab 2 – 示範「Views」)

=====

目的：

- (一)了解 View
- (二)了解 Action Method 回傳值的型別(Type)

=====

步驟：(詳見原文)

(1) 在前例的 TestController 裡面建立一個新的 Action Method 如下：

```
[code]
public ActionResult GetView()
{
    return View("MyView");
}
[/code]
```

(2)建立一個 View，名字叫做「MyView」

方法：把滑鼠游標放在上例的 GetView() 程式上面 按滑鼠右鍵，選 Add View，其他詳見原文，

建立完成後，在 View/Test 資料夾下面會出現一個 MyView.cshtml 的網頁檔，如下圖所示

[img]https://www.codeproject.com/KB/aspnet/866143/lab_1.30.png[/img]

(3) 在 MyView.cshtml 網頁檔裡，鍵入下列程式

```
[code]
[blockquote]
@{
    Layout = null;
}
```

Welcome to MVC 5 Step by Step learning

[/blockquote]

[/code]

(4) 在網址列鍵入：localhost/Test/GetView

則結果如下：

[img]https://www.codeproject.com/KB/aspnet/866143/lab_1.31.png[/img]

=====

討論：

(一) View 的意義：

View 是一個網頁檔，此網頁檔是供 Action Method 來呼叫，做為 Action Method 回應使用者的訊息內容

(二) View 所屬的 Controller：

原則上，大部分的 View 會屬於某個 Controller，
如上例，MyView.cshtml 屬於 TestController

(二) View 所存放的位置：

原則上，View 存放在下列兩類位置中的一類裡面

(1)放在其所屬的 Controller 資料夾下面：

原則上，View 會放在 Views/Contorller Name/ 資料夾下面

Controller Name 為其所屬的 Controller

如上例中，MyView.cshtml 是放在 Views/Test/ 下面

(2)還有一種 View 會放在 Views/Shared 資料夾下面

(三) 那些 Action Method 可以 呼叫 View：

(1)原則上，在某一個 Controller 裡面所有的 Action Method 都可以呼叫 屬於該 Controller 的所有 Views

例如：MyView.cshtml 放在 Views/Test/ 下面

則所有 TestController 裡面的 Action Method 都可以呼叫 MyView.cshtml

所以 GetView() 可以呼叫 MyView.cshtml

(2)不同的 Controller 裡面的 Action Method 不能呼叫 不屬於該 Controller 的所有 Views

如果有一個 Action Method 在 Second Controller 裡面

則它不能呼叫上例中的 MyView.cshtml

但它可以呼叫位於 Views/Second 資料夾裡面的所有的 Views

(3)放在 Views/Shared 的 View 可以被所有任意 Controller 裡面的 Action Method 呼叫

所以 這個資料夾叫 Shared

(四) Action Method 呼叫 View 的方法：

(1)呼叫不同名的 View：

```
[code]
return("View 的名稱", 參數)
[/code]
```

所以上例中的 Action Method GetView() 呼叫 MyView.cshtml 的方法為

```
[code]
return("MyView")
[/code]
```

(2)呼叫同名的 View：

如果 Action Method 與 View 同名，則呼叫時，View 的名稱可省略，如

```
[code]
return View()
[/code]
```

(五) 一個 ActionResult 可以呼叫多個 View，

當然實際上只會呼叫其中的一個

因為一 Return，Method 就結束了

例：

```
[code]
public ActionResult GetView()
{
    if(Some_Condition_Is_Matching)
    {
```

```
return View("MyView");
}
else
{
return View("YourView");
}
}
[/code]
```

(六) Action Method 的傳回值的型別

一、回傳型別可能很多種

在 Lab 1 中，我們看到了 Action Method 可以回傳給使用者的訊息種類很多，所以 Action Method 回傳值的型別可能有很多種

(1)字串型別：

在Lab 1 中，Action Method 我們有回傳 字串的例子，此時，回傳型別為 String

```
[code]
public class TestController : Controller
{
public string GetString()
{
return "Hello World is old now. It's time for wassup bro winking smiley";
}
}
[/code]
```

其中，宣告的地方為

```
[code]
public string GetString()
[/code]
```

所以，回傳型別為 string

(2)物件型別：

在 Lab 1 ，也有回傳物件的例子，此時，回傳值的型別為該物件：

原程式如下：

```
[code]
namespace WebApplication1.Controllers
{
```

```
public class Customer
{
    public string CustomerName { get; set; }
    public string Address { get; set; }
}
public class TestController : Controller
{
    public Customer GetCustomer()
    {
        Customer c = new Customer();
        c.CustomerName = "Customer 1";
        c.Address = "Address1";
        return c;
    }
}
[/code]
```

其中，宣告的地方為

```
[code]
public Customer GetCustomer()
[/code]
```

所以，回傳型別為 Customer 物件

(3) ContentResult

(4) ViewResult

(5) ActionResult

ActionResult 是一個 Abstract Class ,
它有一個子類別(subClass)叫 ViewResultBase ,
然後 ViewResultBase 有一個子類別(subClass)叫 ViewResult

所以 ViewResult 是 ActionResult 下面好幾層的 子類別 (multilevel child)

(6)多型(polymorphism)的回傳值

例：

```
[code]
public ActionResult GetView()
{
    if(Some_Condition_Is_Matching)
```

```
{  
return View("MyView");  
}  
else  
{  
return Content("Hi Welcome");  
}  
}  
[/code]
```

(七) View() Function 的目的

Edited 11 time(s). Last edit at 01/23/2017 02:23AM by RandomVariable.
