[**学习ASP.NET MVC(三)——我的第一个ASP.NET MVC 视图**](http://www.cnblogs.com/chillsrc/p/3727487.html)

今天我将对前一篇文章中的示例进行修改，前一篇文章中并没有用到视图，这次将用到视图。对于前一个示例中的HelloWorldController类进行修改，使用视图模板文件生成HTML响应给浏览器。

**一、创建视图文件**

上一篇文章中的HelloWordController类中的Index方法返回了一串硬编码的字符串消息。而这一次，我将新建一个List方法，用来返回一个View对象，如下面的代码所示：

[复制代码](javascript:void(0);)

public ActionResult List()

{

return View();

}

[复制代码](javascript:void(0);)

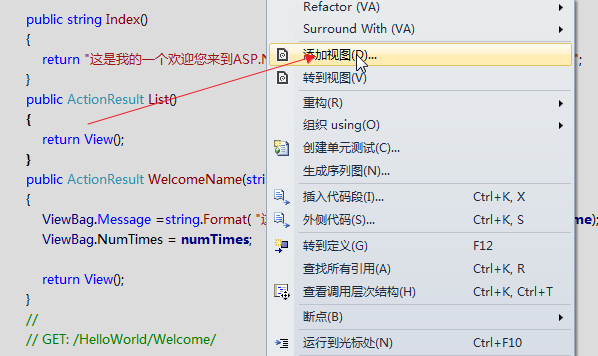
首先，创建一个使用Razor视图引擎的视图模板文件，Razor视图模板文件的扩展名为.cshtml，并提供一种比较优雅的方式使用C＃来创建HTML输出。Razor视图模板减少了编写程序所需要输入的字符数量和敲击键盘的次数，并实现了快速、流畅的编码工作。

如何来创建视图呢？在Visual Studio中为HelloWordController.cs的List控制器方法使用一个视图模板来生成一个HTML的响应给浏览器。

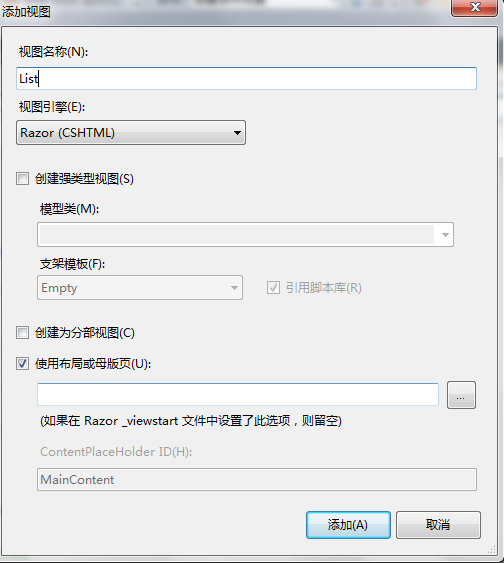
控制器方法（也称为动作方法），如上面的List方法，通常返回一个ActionResult（或是ActionResult的派生类），而不是基本类型，如字符串，整形。   
下面就是添加一个视图方法的具体步骤：

1）先将光标置在List方法中，

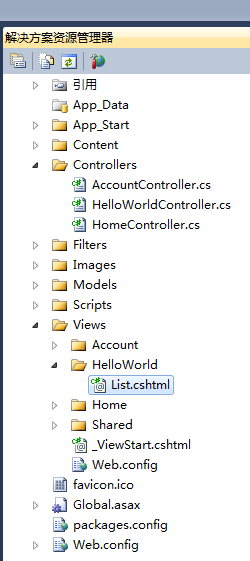
2）单击鼠标右键，Visual studio会弹出一个右键菜单，然后在弹出的右键菜单中，用鼠标左键单击“添加视图”菜单。如下图。



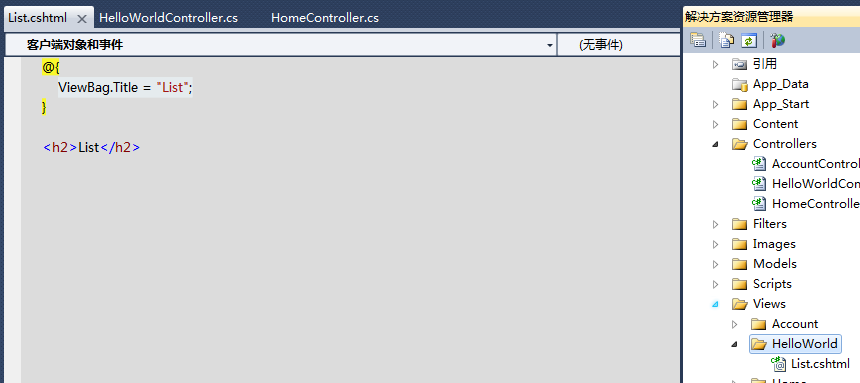
其次，Visual Studio会弹出一个“添加视图”对话框。其中“视图名称”默认为List，其他设置可以修改，本文将保留默认设置，然后点击“添加”按钮：如下图。



这样会在MvcApplication1\Views\HelloWorld文件夹中创建一个List.cshtml文件。如下图。



下图显示了所创建的List.cshtml文件的内容：



添加<H2>HTML标记到以下文件(MvcApplication1\Views\HelloWord\List.cshtml)中。

代码如下。

[复制代码](javascript:void(0);)

@{

ViewBag.Title = "List";

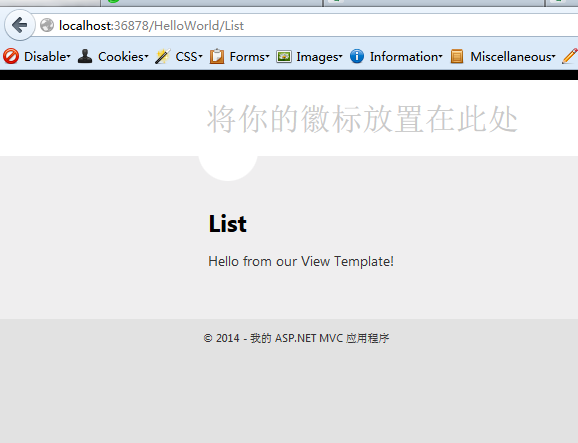
}

<h2>List</h2>

<p>Hello from our View Template!</p>

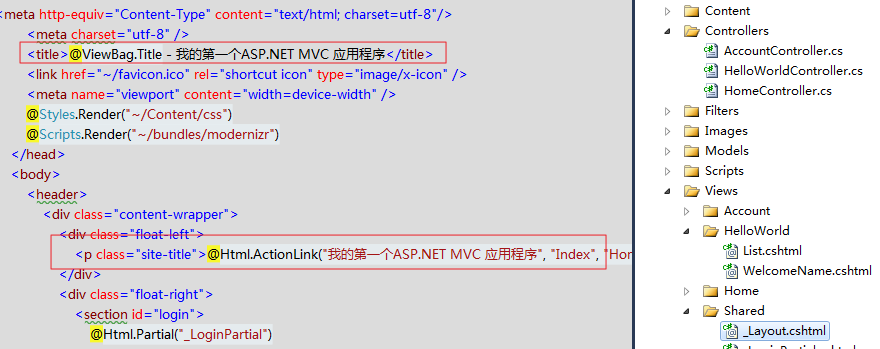
[复制代码](javascript:void(0);)

第三，在Visual studio中按“F5”运行之后，ASP.NET MVC默认调用的是Index方法，而实际需要浏览到HelloWorld控制器的List方法，（如：http://localhost:xxxx/HelloWorld/List）。再来看一下控制器中的List方法，只有一行代码return View()。由于我们没有明确指定的视图文件中使用的名称，ASP.NET MVC默认为使用*\Views\HelloWorld*\List.cshtml视图文件。结果如下图。



程序运行起，效果如上图，看起来还不错。但是，请注意浏览器的标题栏显示“List——我的第一个ASP.NET MVC应用程序”和在页面顶部的“将你的徽标放置在此处”有些不协调，再进行一些改动。

**二、修改视图文件和页面布局**首先，我们要修改在页面顶部的“将你的徽标放置在此处”这个标题。该文本每一页都是相同的，它放在一个共享的视图文件中，这个共享的视图文件又称为布局文件，它是一个公共布局页面，供其他页面使用。在解决方案资源管理器中打开/Views/Shared文件夹中的“\_Layout.cshtml”文件。如下图。



布局模板文件允许你在一个地方填写网站的布局代码之后，可以在多个页面中使用。在布局模板文件中找到@ RenderBody（）行， RenderBody是一个占位符，是用来显示我们所创建特定网页视图。例如，当你在页面中击“关于”链接时，\Home\ About.cshtml视图文件就会显示在RenderBody位置处。如下图中的红色方框部分就是。

  
从“将你的徽标放置在此处”更改布局模板网站标题标题为“我的第一个ASP.NET MVC 应用程序”。

<div class="float-left">

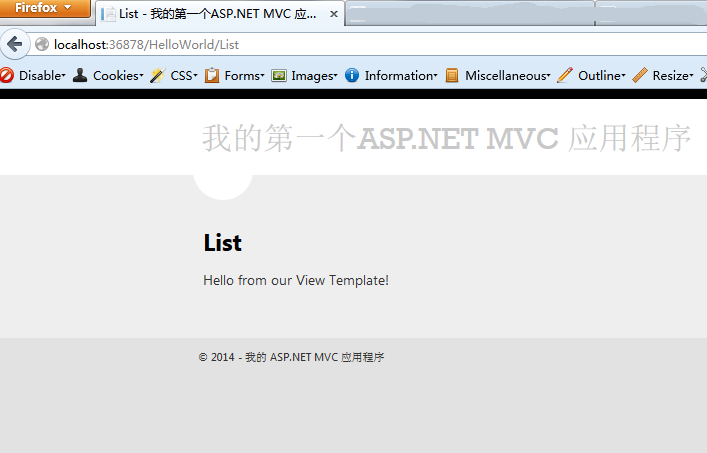
<p class="site-title">@Html.ActionLink("我的第一个ASP.NET MVC 应用程序", "Index", "Home")</p>

</div>

同时修改标题，把标题修改为如下内容：

<title>@ViewBag.Title - 我的第一个ASP.NET MVC 应用程序</title>

运行该应用程序，并注意浏览器中的标题，它会显示为“我的第一个ASP.NET MVC 应用程序”。点击页面上的所有链接，你会发现所有页面的标题都是“我的第一个ASP.NET MVC 应用程序”。这说明，我们只要对布局模板文件进行了修改，那么网站上的所有使用到布局模板的其他页面都会跟着变更。如下图。



现在，让我们改变List 视图的标题。   
打开MvcApplication\Views\ HelloWorld\List.cshtml文件，做以下修改：

第一，   对于页面的标题进行修改

第二，对页面中的二级标头（<h2>元素）进行修改。注意两种的修改需要有点不同，以便能直观的发现，我们修改的是应用程序的哪一部分。

[复制代码](javascript:void(0);)

@{

ViewBag.Title = "ASP.NET MVC 应用程序 List方法";

}

<h2>我的第一个ASP.NET MVC 应用程序 的方法 ——List</h2>

<p>Hello from our View Template!</p>

[复制代码](javascript:void(0);)

“ViewBag.Title = "ASP.NET MVC 应用程序 List方法"”，上面的代码设置List.cshtml视图文件中的ViewBag对象的Title属性。为什么是ViewBag.Title属性呢，你可以打开\_loyout.cshtml文件看看，你会发现在此文件中的<title>元素中会有ViewBag.Title。这样就可以方便的使用这种修改ViewBag属性的方法，轻松修改视图文件和布局文件中的参数。  
运行应用程序并浏览到http://localhost:36878/HelloWorld/list 。请注意，在浏览器中当前页面的标题，次级标题已经改变。 （如果您浏览器中没有发生变化，可能是由于浏览器缓存了之前的内容。请按Ctrl + F5让浏览器强制从服务器加载响应。 ）浏览器页面中显示的标题与我们在List.cshtml文件中设定的ViewBag.Title属性的内容是一样的。如下图中的红色方框所标示的。



上面的示例当中，我们把所有要显示的数据都是硬编码在视图文件中的。虽然这个MVC应用程序有一个“V” （视图） ，有一个“C” （控制器） ，但是却没有“M” （模型）。

**三、从控制器传递数据到视图**

首先，从信息从控制器到视图说起，控制器类通过接收浏览器传入的URL请求，从数据库中检索数据，并最终决定以什么类型的响应发回浏览器，视图负责从一个控制器接收到的数据用于生成和格式化成HTML响应给浏览器。  
控制器负责提供数据或对象给视图模板，让视图模板来根据控制器提供的数据或对象呈现给浏览器。最佳的做法是：一个视图模板不会执行业务逻辑或与数据库直接交互。相反，一个视图模板是否能正确工作是由给它提供数据的控制器所决定的。保持这种“关注点分离”有助于保持代码干净，可测试，更易于维护。

下面的示例是，在HelloWorldController类中写了一个WelcomName方法，这个方法需要两个参数（name与numTimes），然后经过控制器处理之后直接输出结果到浏览器。不是让该控制器直接硬编码为一个字符串输出，而是使用一个视图模板来代替。该视图模板将生成一个动态的响应，这意味着你需要从控制器传递相应的数据给视图，以生成响应。

打开 HelloWorldController.cs文件并且修改WelcomName方法，将NumTimes与name值添加到ViewBag对象。 ViewBag是一个动态对象，这意味着你可以把任何数据类型添加到ViewBag里面。在ASP.NET MVC模型绑定系统会自动从地址栏中获取方法参数的字符串，并映射命名参数（name和numTimes ） 。

[复制代码](javascript:void(0);)

public ActionResult WelcomeName(string name, int numTimes = 1)

{

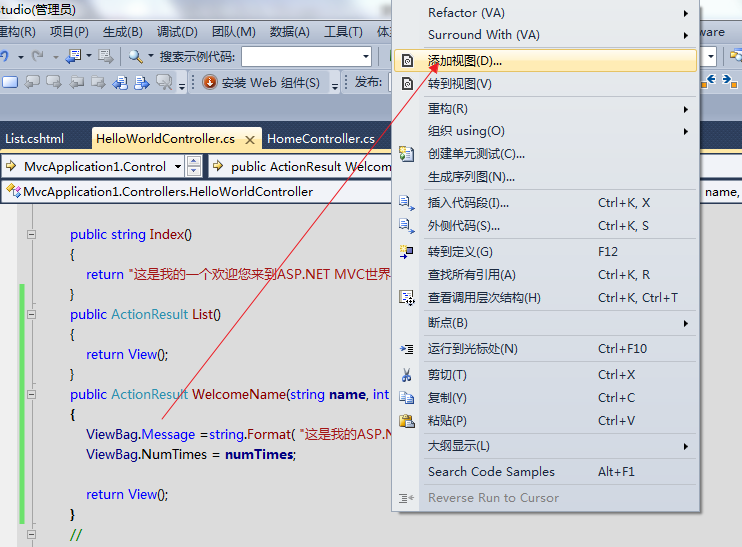
ViewBag.Message = "Hello " + name;

ViewBag.NumTimes = numTimes;

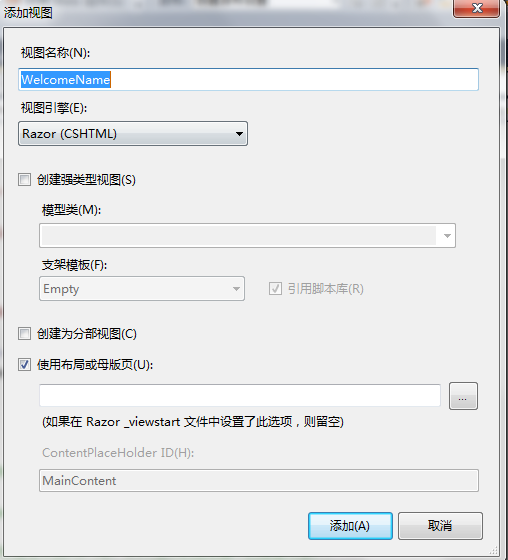
return View();

}

[复制代码](javascript:void(0);)

上面的代码中把要传递给视图的数据包含在了ViewBag对象中了。   
然后，你需要一个WelcomeName视图模板​​！如下图，新建一个新的视图文件。   


Visual Studio会弹出一个“添加视图”对话框。按下图所示，填写相应数据。



然后单击“添加”按钮，Visual Studio会添加一个新的WelcomeName.cshtml文件，在此文件的<h2>元素的下方添加以下代码。下面的完整WelcomeName.cshtml文件如下所示。

[复制代码](javascript:void(0);)

@{

ViewBag.Title = "这是我的ASP.NET MVC世界的欢迎WelcomeName";

}

<h2>WelcomeName</h2>

<ul>

@for (int i=0; i < ViewBag.NumTimes; i++) {

<li>@ViewBag.Message</li>

}

</ul>

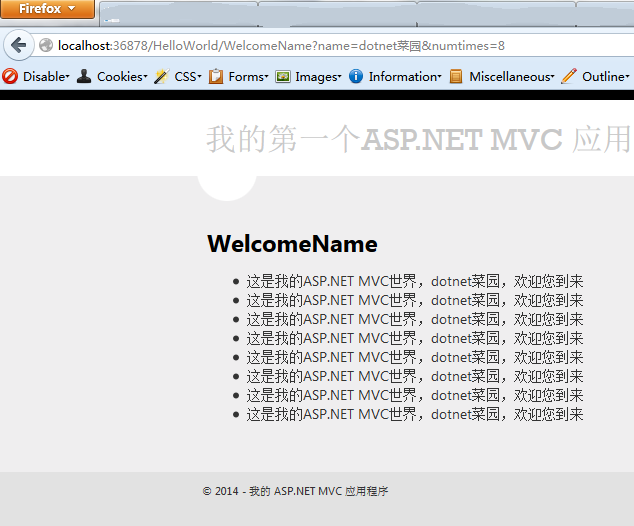
[复制代码](javascript:void(0);)

1）运行应用程序并使用浏览器浏览以下网址：   
http://localhost:xx/HelloWorld/WelcomeName?name=dotnet菜园&numtimes=8  
2）URL中的数据是先从URL中传递给指定的控制器

3）指定的控制器将数据打包成一个ViewBag对象

4）控制器再把打包好的对象传递给视图，然后该视图显示数据给用户看。你会看到八条欢迎信息。

如下图。



关于ASP.NET MVC中的视图部分，就学习到这里，下次再继续。