初识MVC

目标：了解MVC原理，构建带模板的MVC项目。

重点：MVC架构模式、请求响应过程

难点：适应页面请求转控制器请求模式变化和语法表达

# ASP.NET与MVC

扩展阅读：<https://docs.microsoft.com/zh-cn/aspnet/overview>

ASP.NET 是一个用于构建网站和 web 应用程序web 框架。

MVC是组织和管理代码的一种架构体

MVC不专属于ASP.NET，ASP.NET MVC专指基于ASP.NET的MVC

## ASP.NET的历史主流架构—webform

扩展阅读：

https://docs.microsoft.com/zh-cn/aspnet/web-forms/index

## ASP.NET的替代方案—ASP.NET CORE

扩展阅读：

<https://docs.microsoft.com/zh-cn/aspnet/core/?view=aspnetcore-2.2>

## ASP.NET的当前应用架构—MVC

MVC扩展阅读

<https://docs.microsoft.com/zh-cn/aspnet/mvc/overview/index>

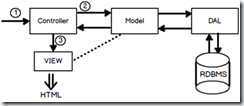
# ASP.NET MVC

## 请求流的通用步骤

Step 1：首先获取控制器。

Step 2：依赖行为控制器创建Model对象，Model通过转换调用数据访问层。

Step 3：数据填充Model之后，传递到View 显示层，实现显示的目的。

[](http://images0.cnblogs.com/blog/139239/201506/121203500192900.png)

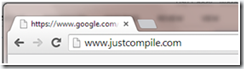
## 控制器Controller

在Asp.Net MVC中，C代表Controller，用于**处理用户交互逻辑**的。

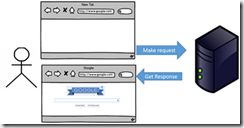
### 什么是用户交互逻辑？

**场景1**

当用户输入URL摁下回车键时，会发生什么事情？

[](http://images0.cnblogs.com/blog/139239/201506/121203507235285.png)

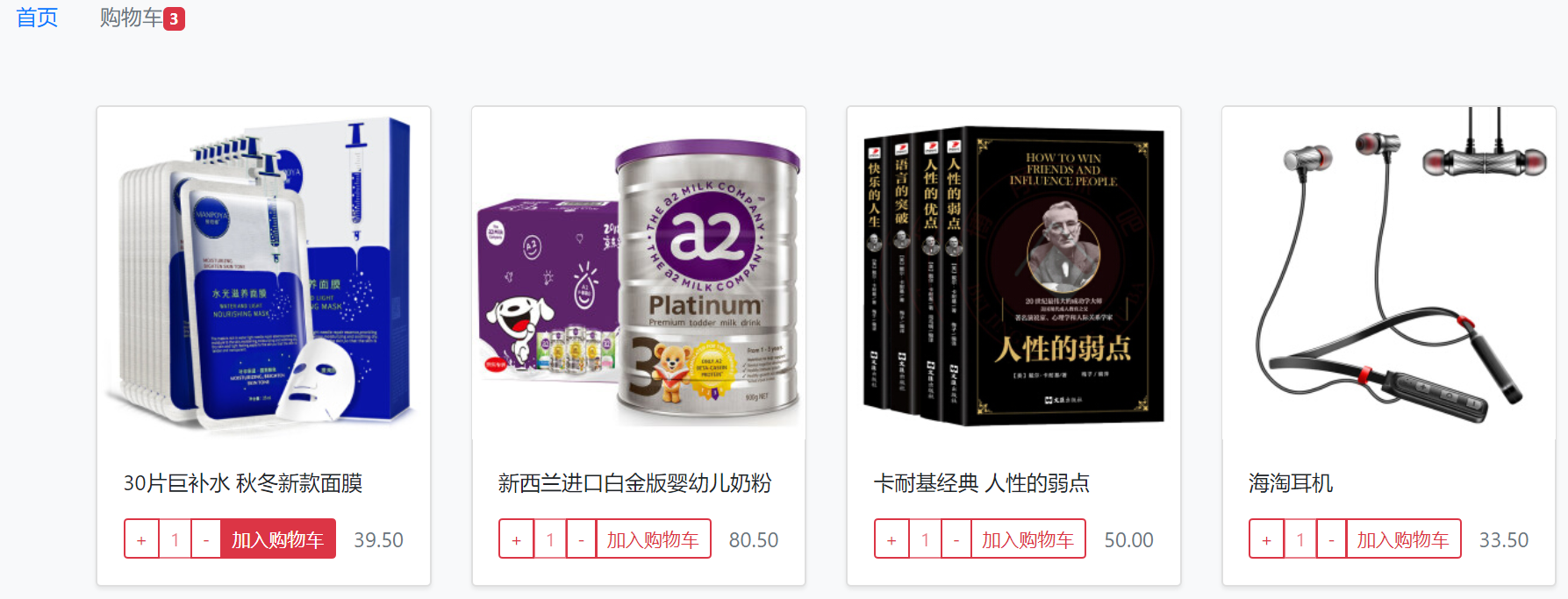
浏览器首先需要给服务器发送请求，服务器再做出响应。

[](http://images0.cnblogs.com/blog/139239/201506/121203517541870.png)

通过这些请求之后，客户端正尝试与服务器交互，服务器能够反馈响应，因为服务器端存在一些判断逻辑来处理这些请求。**这些能够处理用户请求以及用户交互行为的业务逻辑称为用户交互逻辑。**

**场景2**

有一种常见的情况，服务器端发送的请求是HTML请求。HTML请求是由一组输入控件和提交按钮组成的。



当用户点击“加入购物车”按钮之后会发生什么？

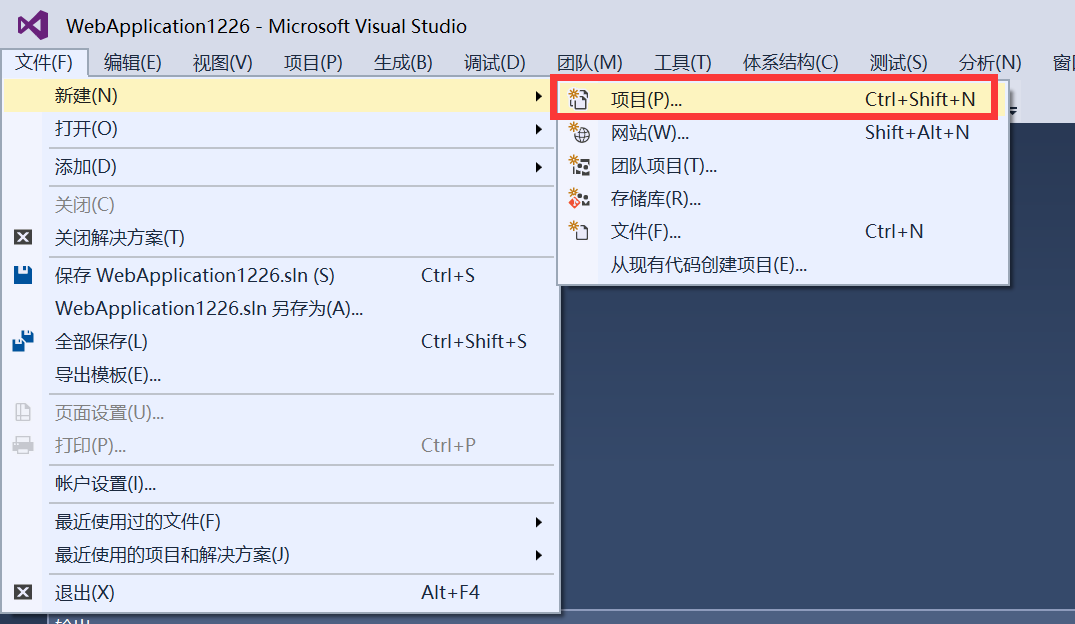
客户端编程—触发单击事件

服务器端编程—向服务器端发送一个简单的HTTP请求。服务器端用相应的逻辑，响应请求。简单来说是将用户交互逻辑写在服务器端。

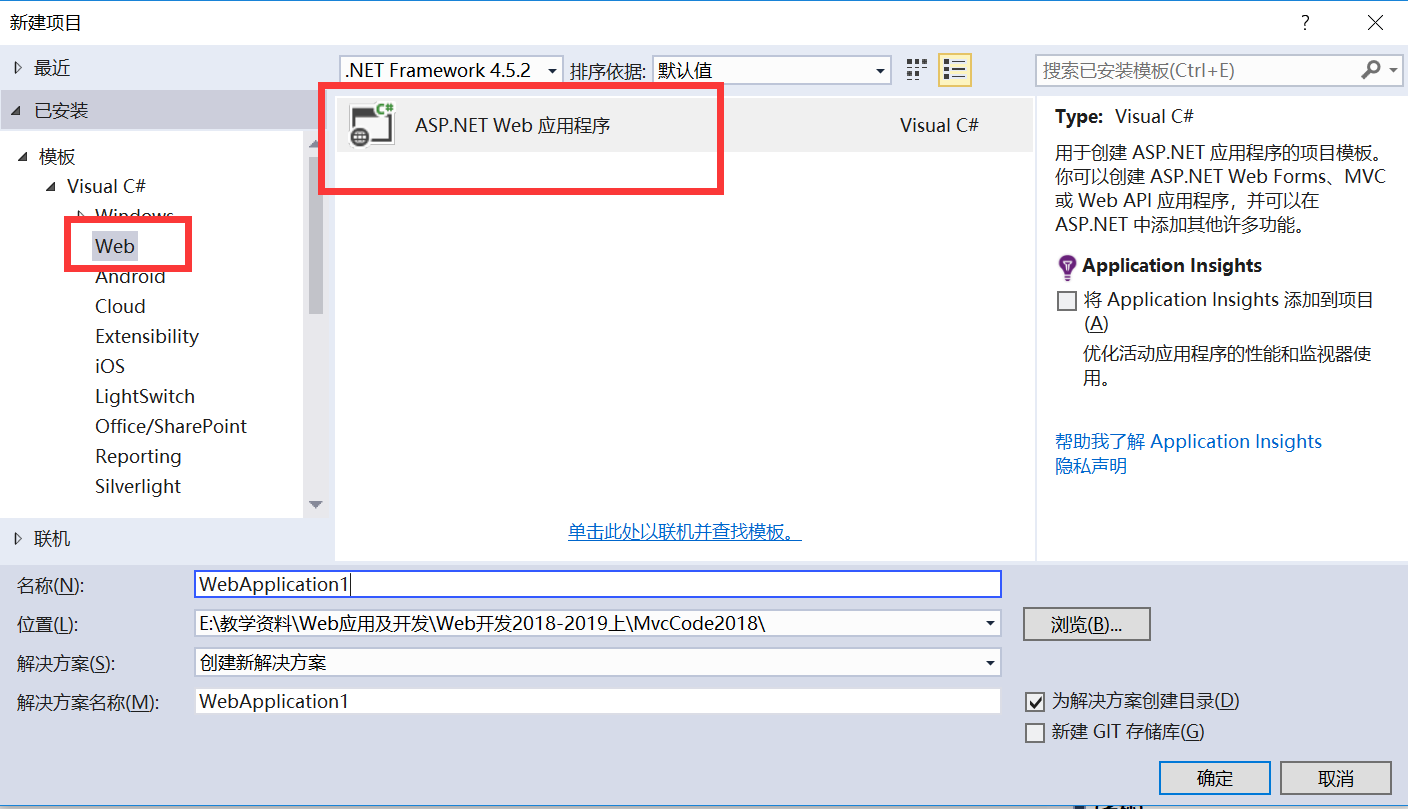
### 实验一 MVC Hello world—学习控制器

* **Step1 创建一个Asp.Net MVC 5项目**

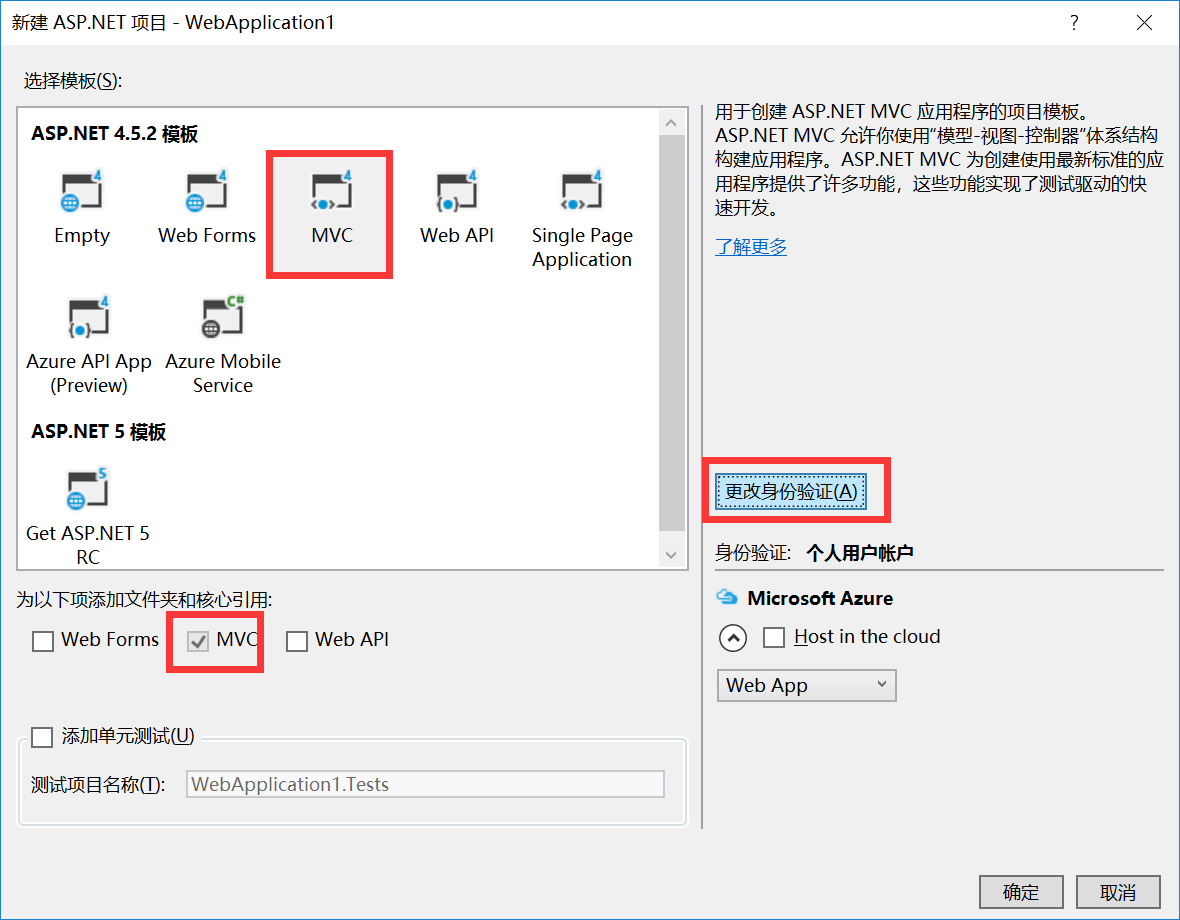
打开Visual studio 点“文件”->新建->项目。



* Step 1.2 选择Web 应用，输入项目名称，选择存放路径，点击确定。



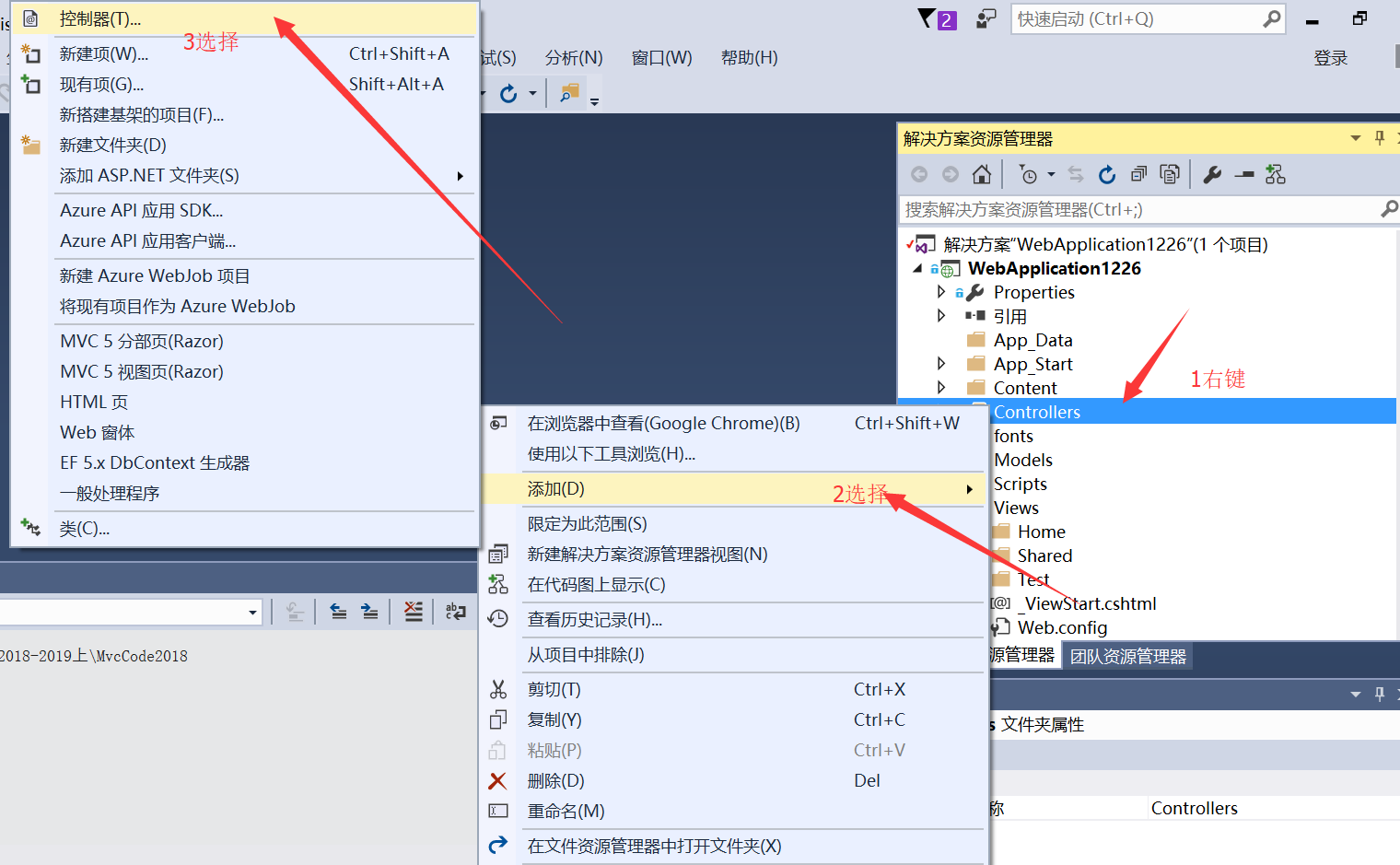
* Step 1.3 选择MVC 模板



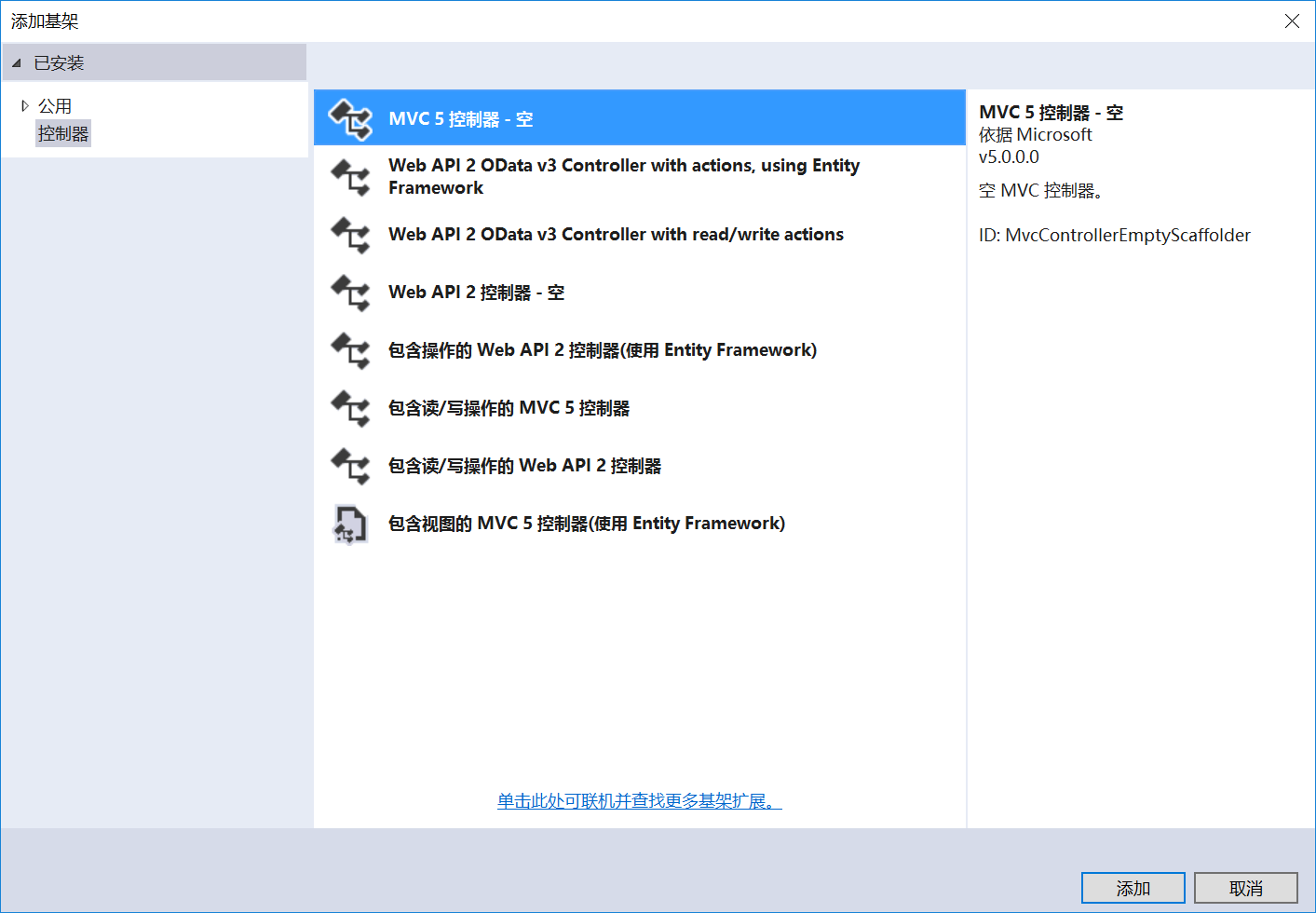
* Step 1.4 选择更改身份验证，弹出对话框中选择“不进行身份验证”，并点击确定。



* Step 2 –创建控制器
* Step 2.1，在资源管理器中，右击controller文件夹，选择添加->Controller(控制器)

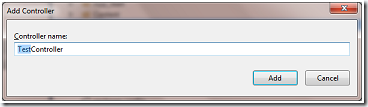


* Step 2.2 选择空 MVC 5 Controller 并点击添加



* Step 2.3 输入控制器的名称”TestController“，点击添加。

在这一步骤中，要特别注意千万不能删除名称中的” Controller”关键字。名称中必须包含Controller关键字。

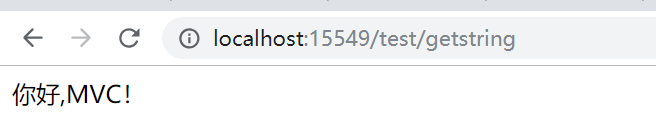
[](http://images0.cnblogs.com/blog/139239/201506/121204001913893.png)

* Step 3. 创建行为方法

打开新建的TestController 类，可以发现已生成的Index 方法，将该方法删除，并且添加新方法命名为GetString ，代码如下：

|  |
| --- |
| public class TestController : Controller  { public string GetString()  {  return "你好,MVC！";  }  } |

Step 4. 运行并测试 按 F5 键，在地址栏中以“ControllerName/ActionName”这样的形式输入，需要注意的输入控制器名称时，不能输入”Controller“只输入”Test”。



### 实验一：Q&A

1. TestController 和Test之间的关系是什么？

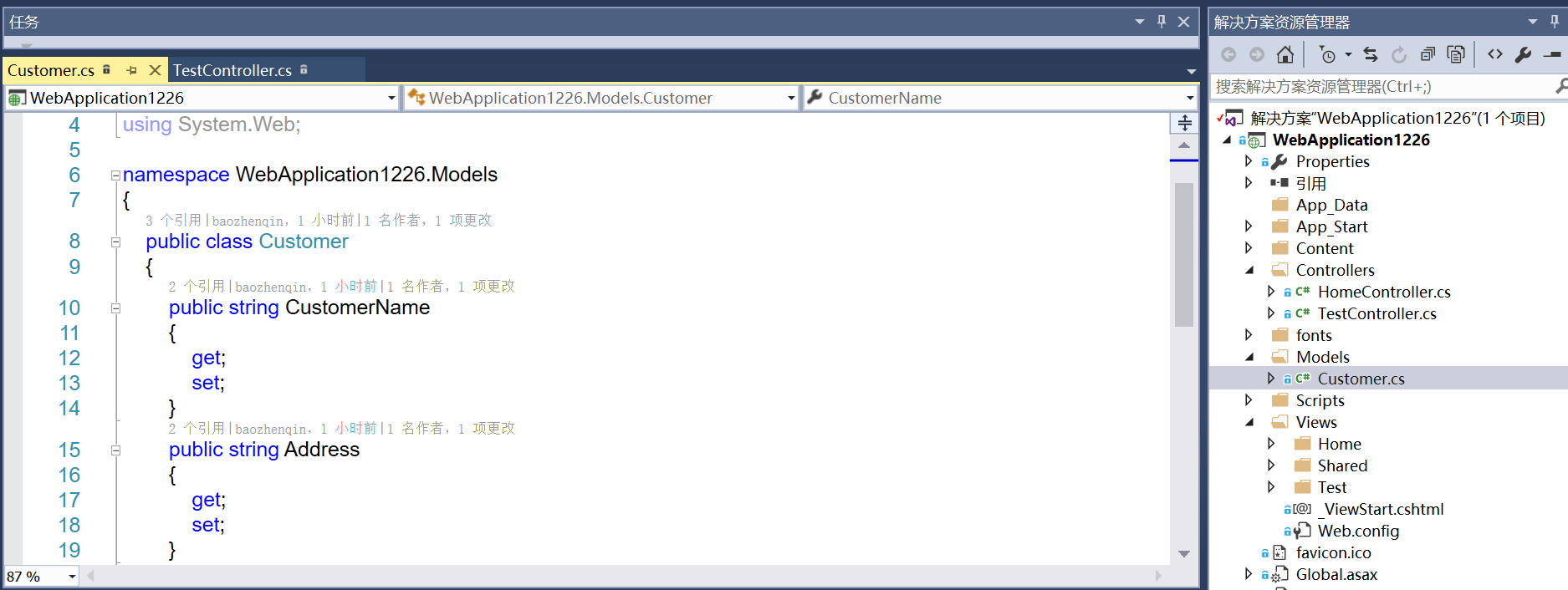
TestController是类名称，而Test是Controller的名称，请注意，当你在URL中输入controller的名称，不需要输入Controller这个单词。

2. Action（行为） 方法是什么？

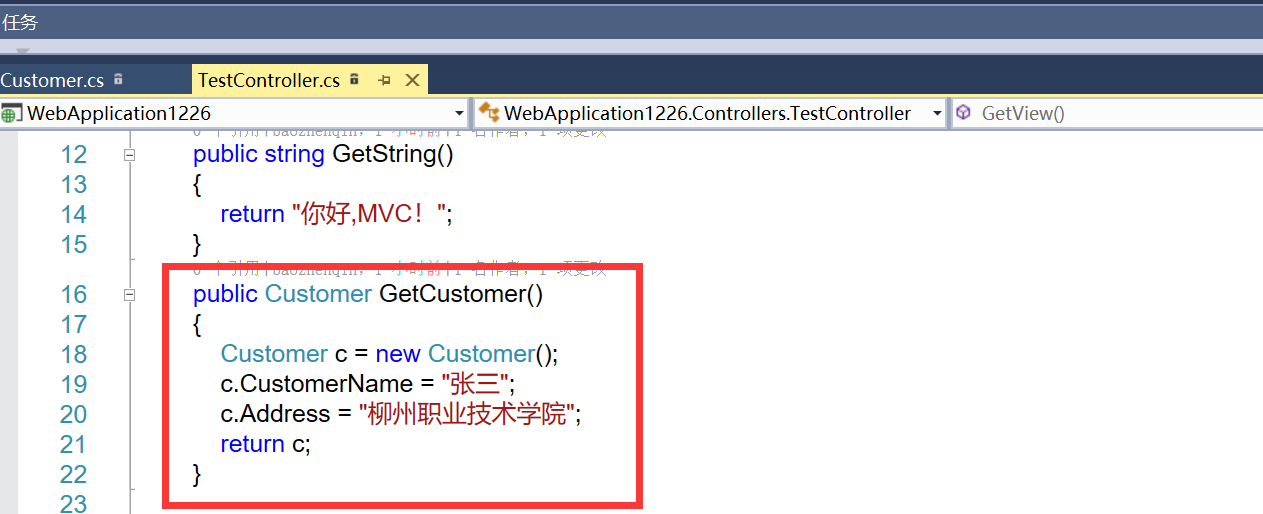
Action 方法 简单的来说就是一个Controller内置的public类型的方法，能够接收并处理用户的请求，上例中，GetString 方法返回了一个字符串类型的响应。

3. 如果从Action 方法中返回对象值会出现什么意外情况？

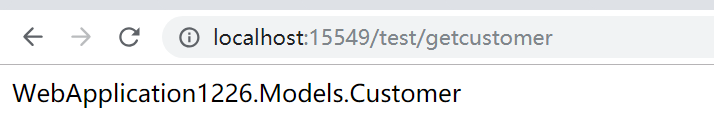
（1）在Models文件夹中添加一个类Customer,并设置两个属性 CustomerName和Address



（2）在Test控制器中增加一个Action



（3）运行程序并定位到http://localhost:15549/test/getcustomer



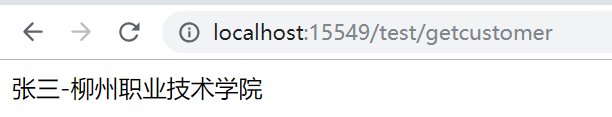
当返回类型如“Customer”这样类似的对象时，将调用ToString()方法，返回“NameSpace.ClassName”形式的类名。

4. 如果需要获得上面例子中的属性值，要如何操作？

简单重写类的“ToString”方法，如下：

|  |
| --- |
| public override string ToString()  {  return this.CustomerName + "-" + this.Address;  } |

运行结果：



5. Action 方法是否只能用Public修饰符来修饰？

答案是肯定的，每个公有方法都会自动称为Action 方法。

6. 非public方法是什么？

类的方法都比较简单，并且并不是公共可用的。无法在Web中调用。

7. 如果我们需要其他函数来完成一些特定功能，但不是Action Method要如何实现？

使用NonAction属性修饰，如下：

[NonAction]

public string SimpleMethod()

{

return "Hi, I am not action method";

}

当尝试给以上Action 方法发送请求时，会获得以下结果：



## View部分

Controller是处理用户请求，并做出响应，通常情况下响应都是以显示在浏览器中，使用HTML代码，浏览器才可识别。HTML有图像，文本，输入控件等。通常称为用户界面的设计即UI层，在ASP.net MVC称为View。

### 实验二——深入理解View

在实验二中，创建一个简单的MVC应用，仅仅具有Controller和简单的字符串类型的返回值。

Step1 –创建新的Action 方法

在TestController中添加新的Action 方法，如下：

public ActionResult GetView()

{

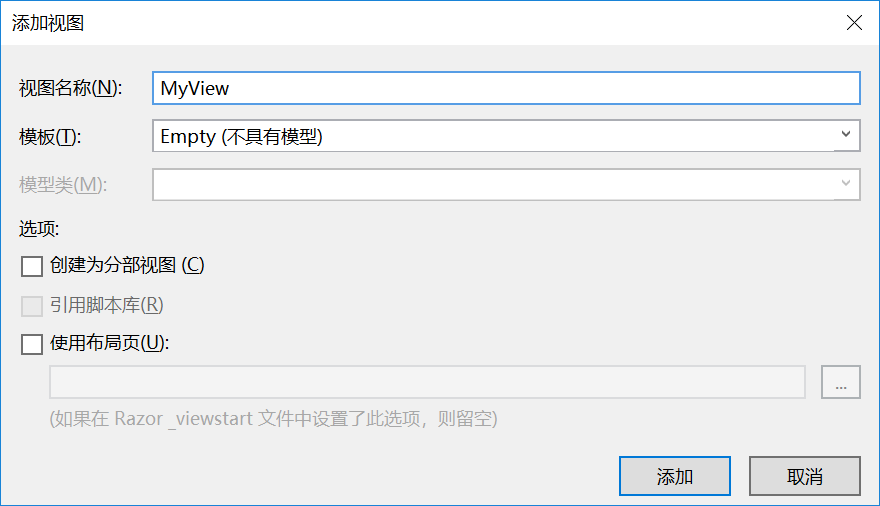
return View("MyView");

}

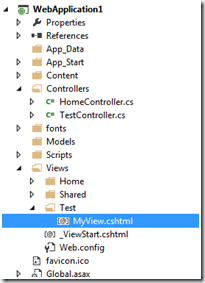
* Step 2 创建View
* Step 2.1 右击上述创建的Action 方法，选择“添加View”



* Step 2.2 在添加View的对话框中输入View名称“MyView”，取消选择“使用布局”的复选框，点击添加。



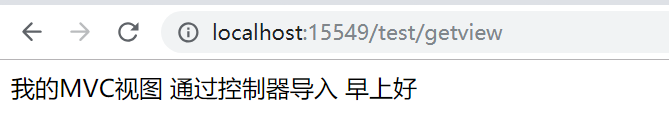
资源管理器重的Views/Test文件夹中会添加一个新的View文件。

[](http://images0.cnblogs.com/blog/139239/201506/121204075826102.png)

* Step3 在View中添加内容

打开MyView.cshtml 文件，并添加以下内容：

* Step 4. 运行 按F5键运行应用



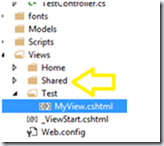
* **实验二：Q&A**

1. 为什么View会放在Test的文件夹中？

View是与放置在特定目录下的Controller相关。这个特定文件夹是以”ControllerName”命名的，并且放在View文件夹内

2. 在多个控制器中无法重用View吗？

当然可以，我们需要在将这些文件放在特定的Shared文件夹中。将View 放在Shared文件夹中所有的Controller都可用。

[](http://images0.cnblogs.com/blog/139239/201506/121204091768114.png)

3. 单个Action 方法中可引用多个View吗？

可以，ASP.NET MVC的view和Controller不是严格的匹配的，一个Action Method可以引用多个view，而一个View也可以被一个Action方法使用如下代码所示：

1: public ActionResult GetView()

2: {

3: if(Some\_Condition\_Is\_Matching)

4: {

5: return View("MyView");

6: }

7: else

8: {

9: return View("YourView");

10: }

11: }

4. View函数的功能是什么？

 创建 ViewResult 对象将会渲染成视图来给用户反馈

* ViewResult 创建了ViewPageActivator 对象
* ViewResult 选择了正确的ViewEngine，并且会给ViewEngine的构造函数传ViewPageActivator对象的参数
* ViewEngine 创建View类的对象
* ViewEngine 调用View的RenderView 方法。

5. ActionResult和 ViewResult的关系是什么？

ActionResult是抽象类，而ViewResult是ActionResult的多级孩子节点，多级是因为ViewResult是ViewResultBase的子类，而ViewResultBase是ActionResult的孩子节点。

6. 什么是ContentResult？

ViewResult是HTML响应而ContentResult是标准的文本响应，仅返回字符串类型。区别就在于ContentResult是ActionResult的子类。

## MVC项目文件组织结构

（与WEBForm比较）

|  |  |
| --- | --- |
| MVC |  |
|  | 控制器：Controllers  视图：Views  配置文件：除web.config之外，各种配置文件进App\_Star文件夹 |