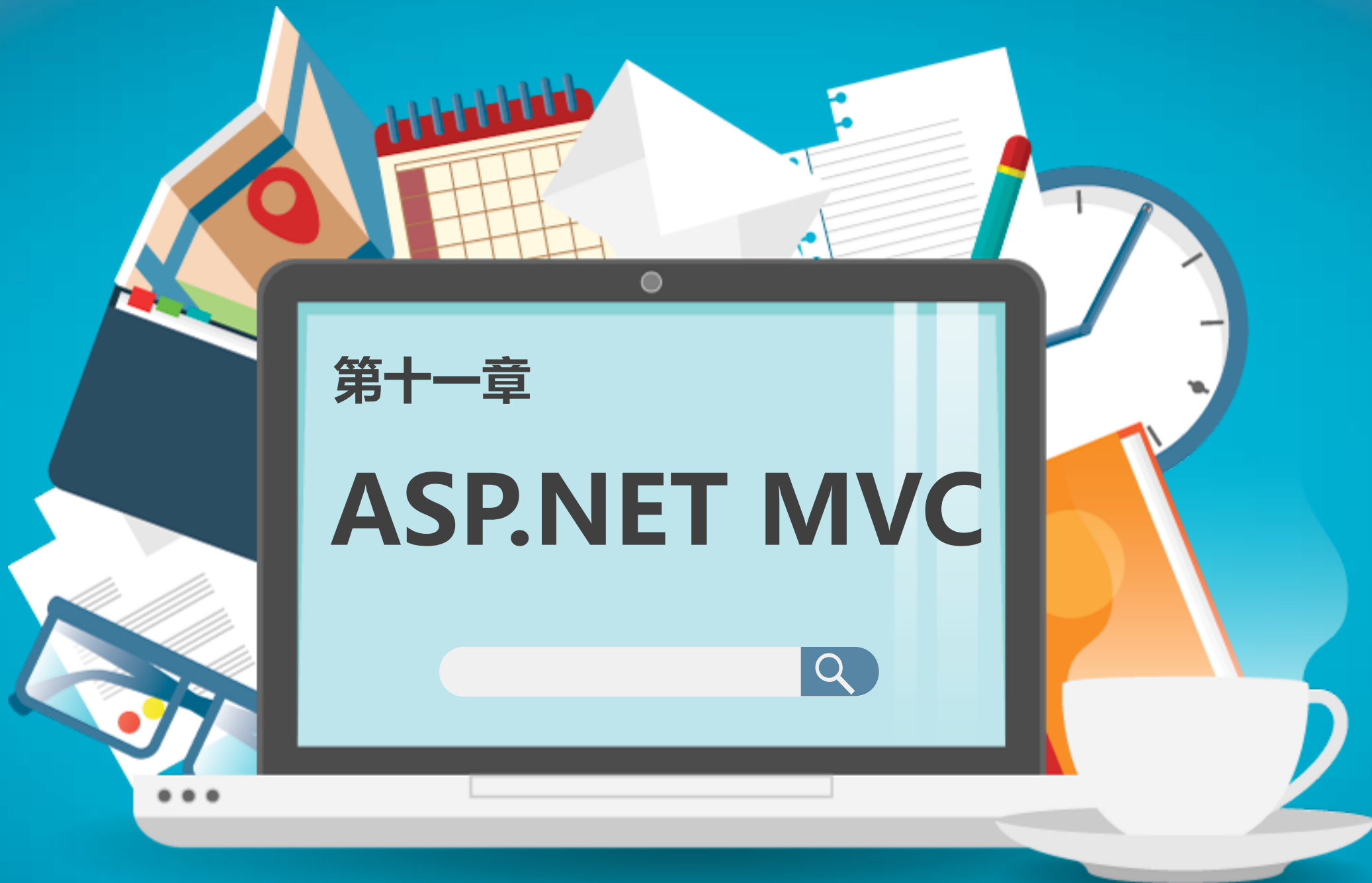
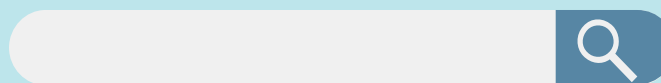


第十一章

ASP.NET MVC





一、ASP.NET MVC创建



二、Visual studio安装



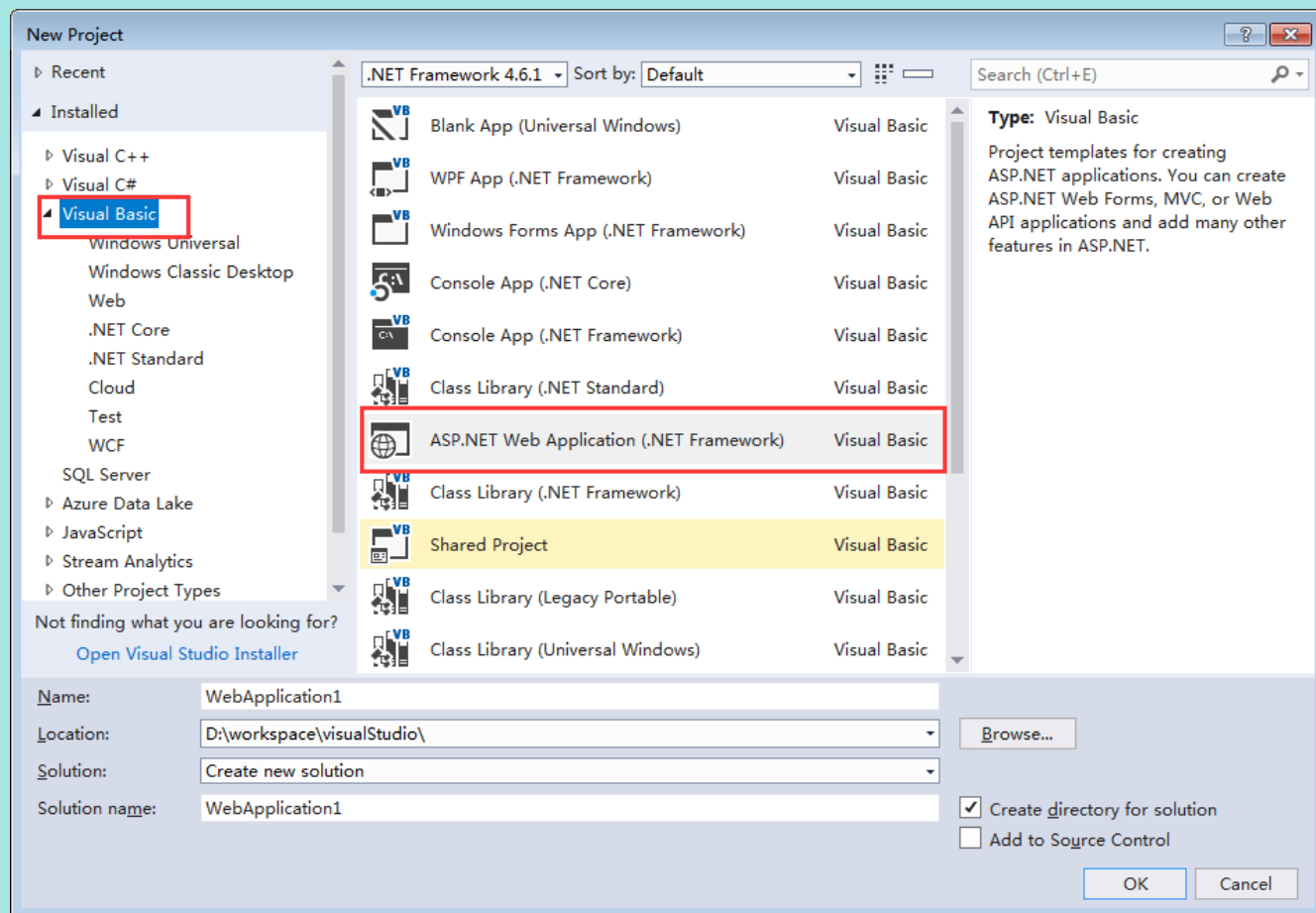
三、Hello world



四、练习




一、ASP.NET MVC创建





一、ASP.NET MVC创建





New ASP.NET Web Application - Lesson11-1


Empty


Web Forms


MVC


Web API


Single Page Application

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use [Learn more](#)

[Change Authentication](#)

Authentication: **No Authentication**

Add folders and core references for:

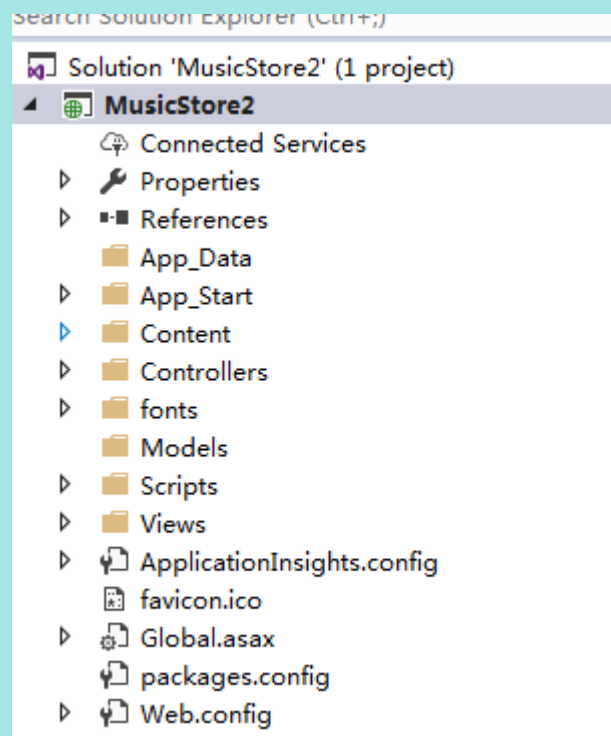
☐ Web Forms ☒ MVC ☐ Web API

☐ Add unit tests

Test project name:

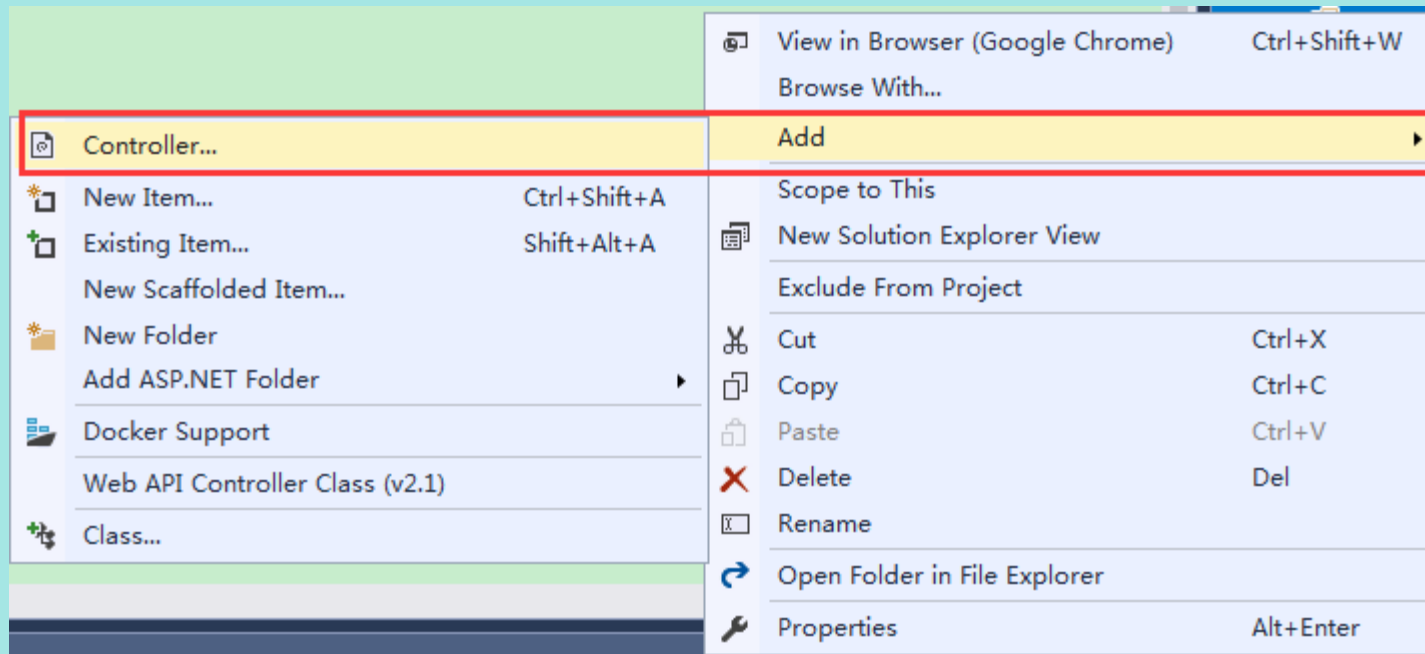
OK Cancel

一、ASP.NET MVC创建



/Controllers	Controllers响应浏览器的输入，决定用这些输入来做什么，并且返回response给用户
/Views	Views用来放UI模板
/Models	Models用来保持和操作数据
/Content	Content来放images，CSS等静态资源
/Scripts	放JavaScripts文件

1.1Controller



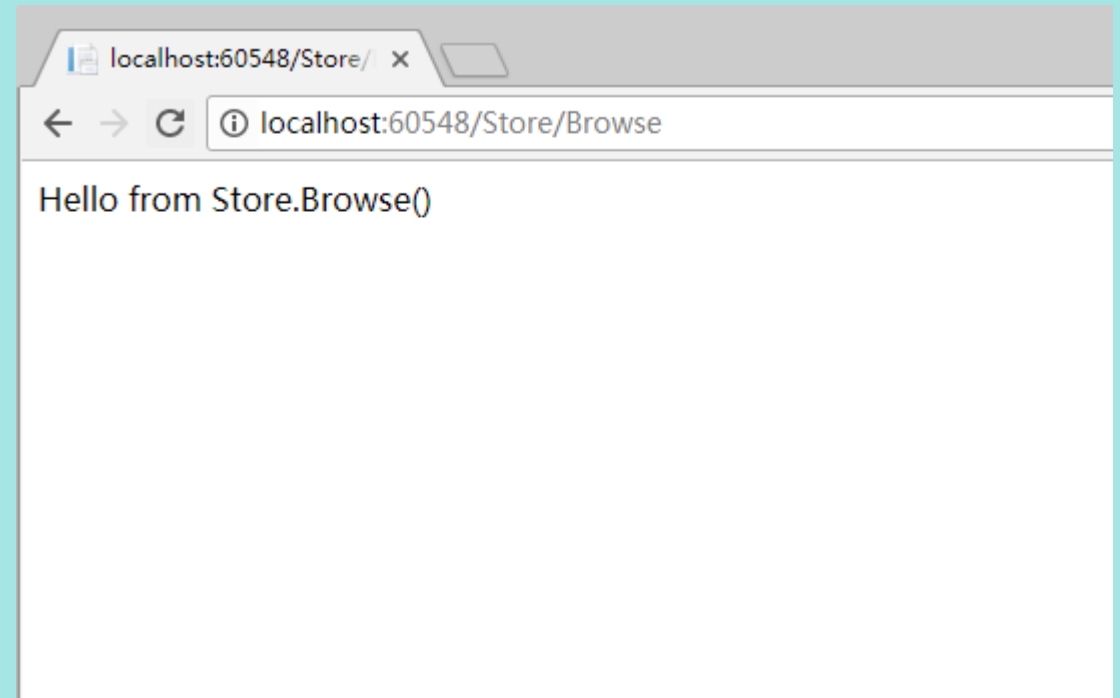
1.1Controller



```
Public Class HomeController
    Inherits
System.Web.Mvc.Controller
    Function Index() As ActionResult
        Return View()
    End Function
End Class
```

让我们再建一个StoreController来返回一个简单的字符串看一下效果

```
' GET: Browse
Function Browse() As String
    Return "Hello from Store.Browse()"
End Function
```



1.1Controller



让我们来接收URL的querystring，并做一些简单的处理返回给浏览器。

' GET: Browse

```
Function Browse(genre As String) As String
```

```
    Dim message As String =  
HttpUtility.HtmlEncode("Store.Browse,  
Genre = " & genre)
```

```
    Return message
```

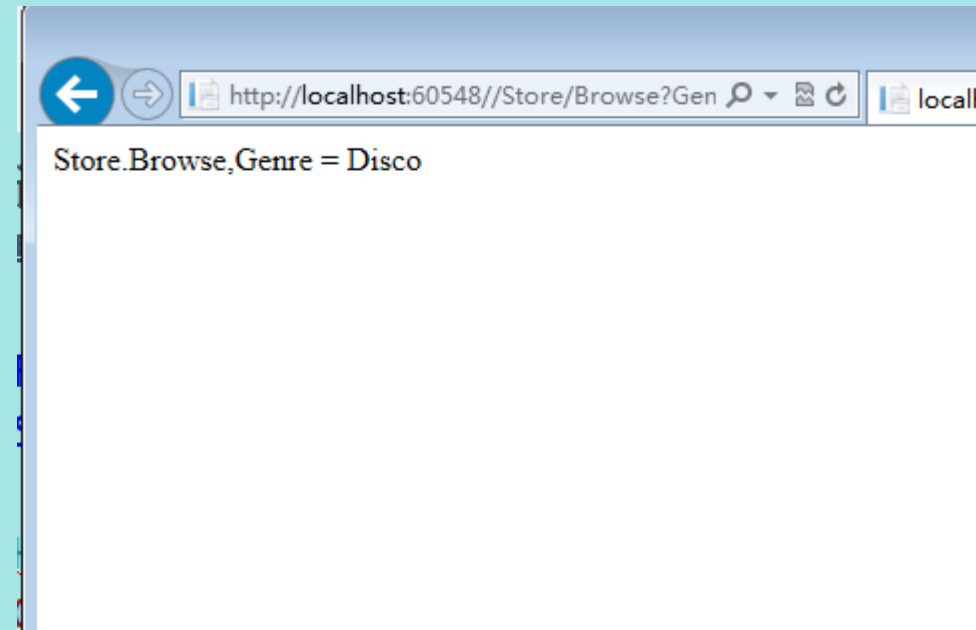
```
End Function
```

/Store/Browse?Genre=Disco

注意：我们用HttpUtility.HtmlEncode来“净化”用户的输入，阻止用户注入JavaScript到我们的view中（

/Store/Browse?Genre=<script>>window.location=' http://hackersite.com' </script>.

）新版.net不允许传html标签

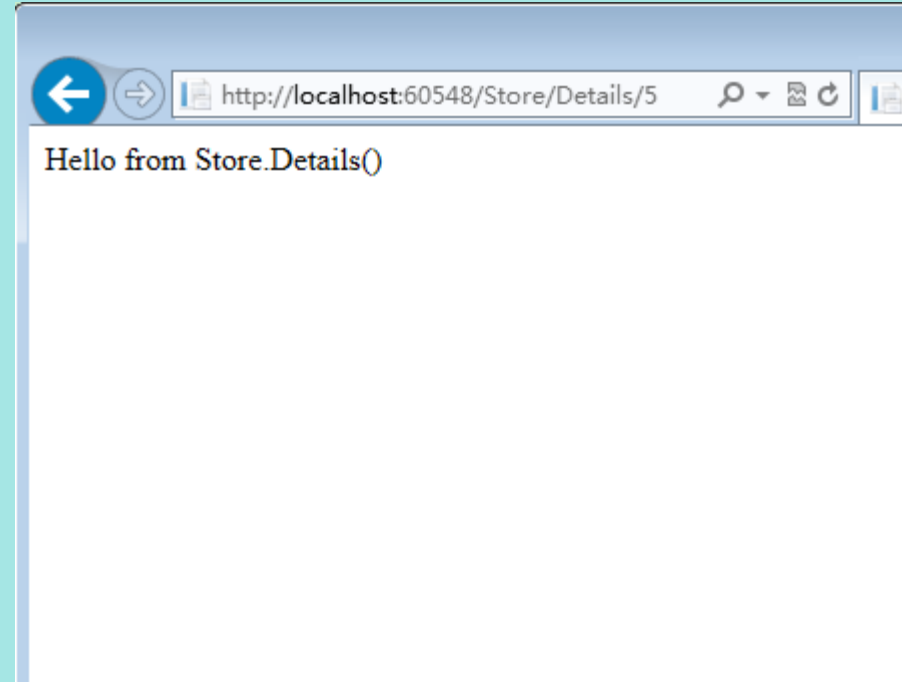


1.1Controller



ASP.NET可以自动转换RestFull风格的URL，并接收其中的参数，并且不需要任何的配置

```
' GET: Details /Store/Details/5
Function Details(id As Integer) As String
    Dim message = "Store.Details, ID = " & id
    Return "Hello from Store.Details()"
End Function
```



1.1Controller



Controller的可选参数

```
Function WelCome(ByVal name As String, Optional ByVal times As Integer = 1) As String
    Dim msg As String = "欢迎" & name & ", 访问" & times & "次"
    Return "" & Server.HtmlEncode(msg) & ""
End Function
```

← → ↻ ⓘ localhost:54712/HelloWord/WelCome?name=%27小王%27

欢迎'小王', 访问1次

← → ↻ ⓘ localhost:54712/HelloWord/WelCome?name=%27小王%27×=2

欢迎'小王', 访问2次

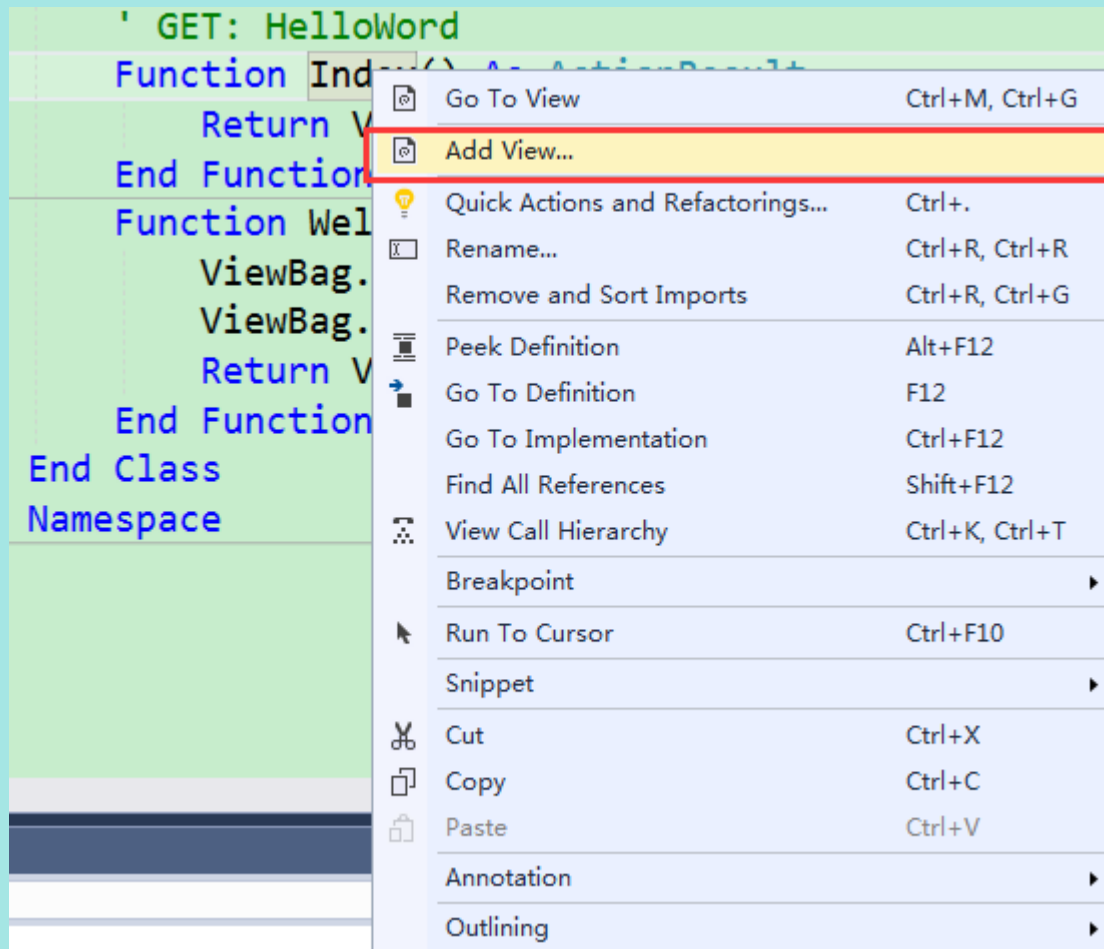
1.1Controller



为了使用view-template,HomeController的
返回值为ActionResult,并让他返回View()

```
Public Class HomeController
    Inherits
System.Web.Mvc.Controller
    Function Index() As ActionResult
        Return View()
    End Function
End Class
```

1.2View



1.2View



View使用Razor语法

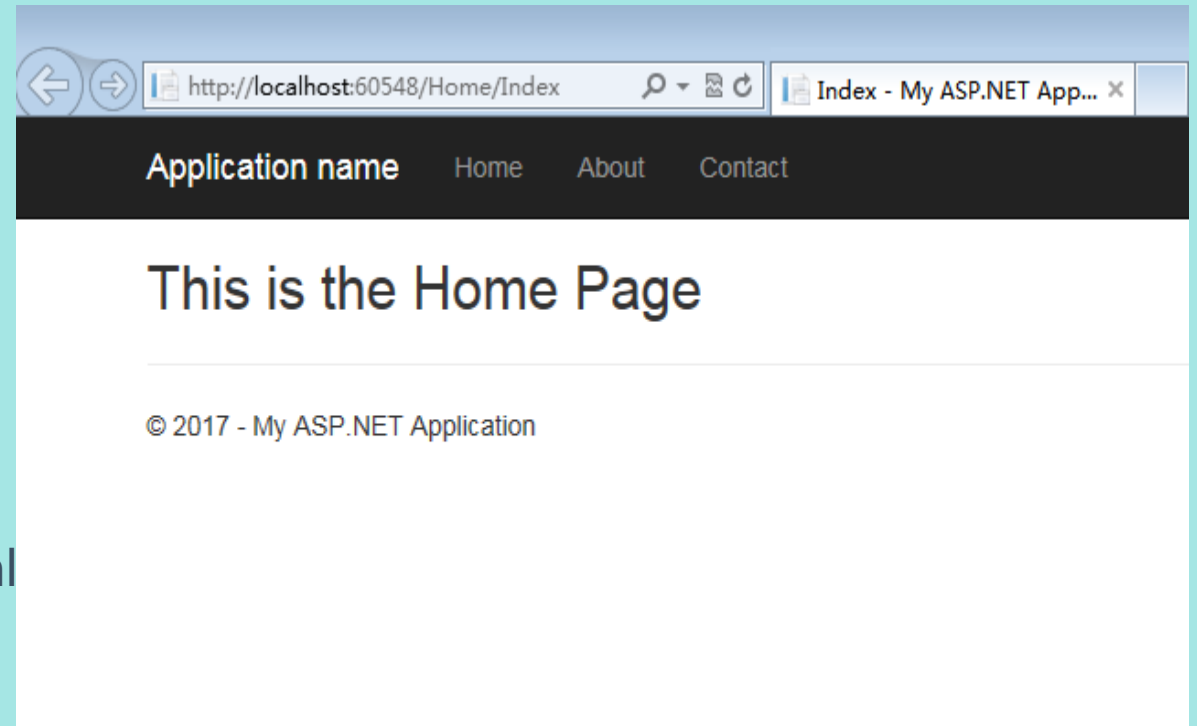
@Code

```
    ViewData("Title") = "Index"
```

End Code

```
<h2>This is the Home Page</h2>
```

大多数网站都会有很多公共的内容，比如导航栏，底部，logo，公用的样式等等，Razor view用views/shared/_Layout.vbhtml来做公共内容然后通过@RenderBody()来引入每个view的内容



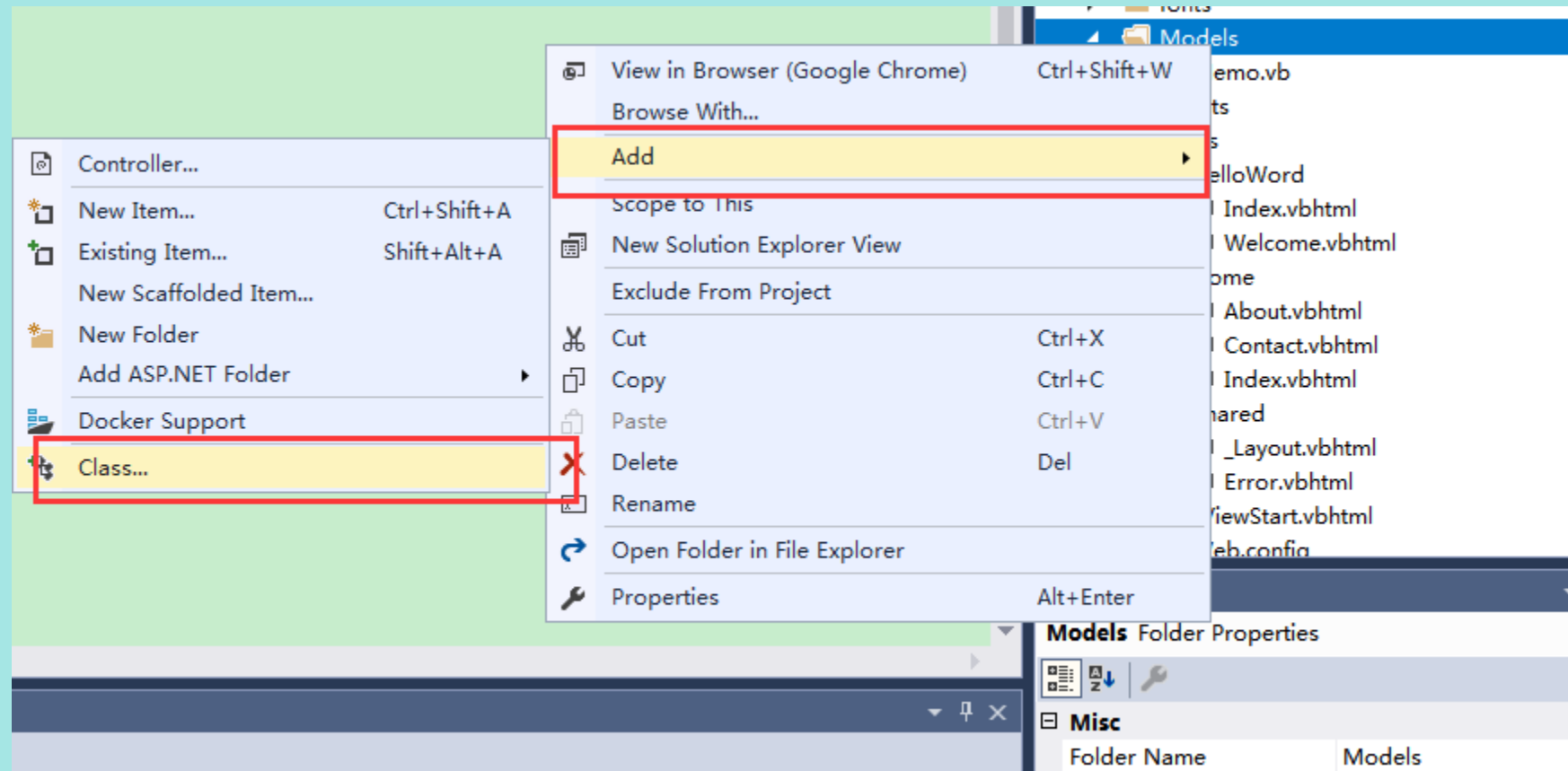
1.3 Model



使用Model向View传递信息

HTML展示的都是一些静态消息，为了建立一个动态网站，我们就需要向template传递数据
当Controller的返回值为ActionResult的时候，可以把Model传递给View

1.3 Model



1.3Model



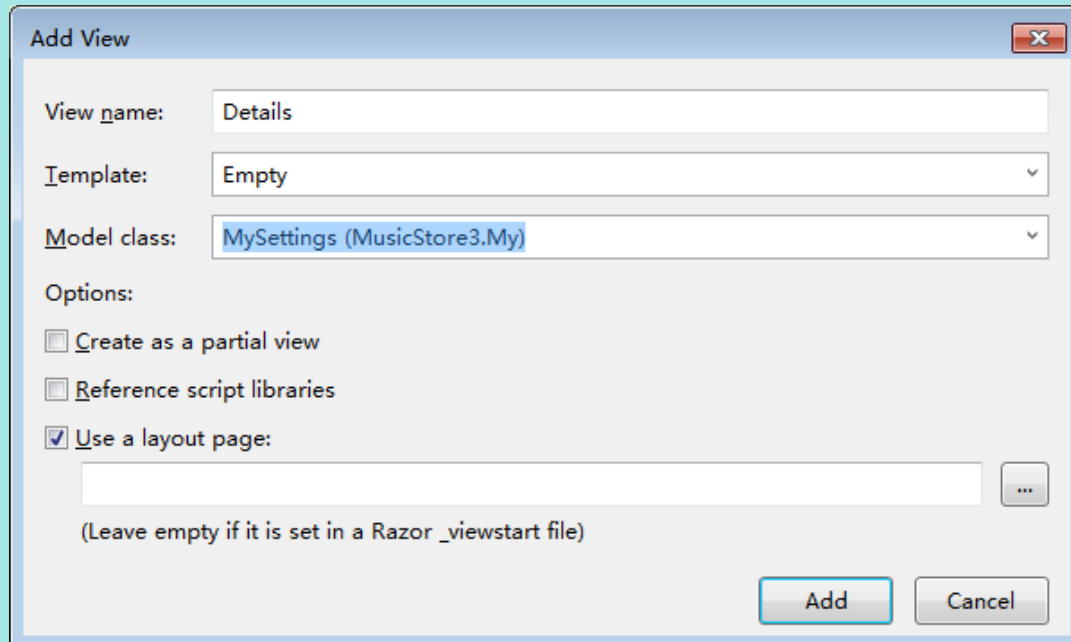
```
Public Class Genre
    Public Property Name As String
End Class
Public Class Album
    Public Property Title As String
    Public Property Genre As Genre
End Class
```

Model会自动给属性填充get和set方法

1.3Model



```
' GET: Details /Store/Details/5
Function Details(id As Integer) As ActionResult
    Dim album As Album = New Album()
    album.Title = "Album" & id
    Return View(album)
End Function
```



The 'Add View' dialog box is shown with the following configuration:

- View name:** Details
- Template:** Empty
- Model class:** MySettings (MusicStore3.My)
- Options:**
 - ☐ Create as a partial view
 - ☐ Reference script libraries
 - ☒ Use a layout page:

Below the 'Use a layout page' option is an empty text field with a browse button (...). A note below the field states: (Leave empty if it is set in a Razor _viewstart file).

At the bottom are 'Add' and 'Cancel' buttons.

1.3Model



```
' GET: Details /Store/Details/5
Function Details(id As Integer) As ActionResult
    Dim album As Album = New Album()
    album.Title = "Album" & id
    Return View(album)
```

End Function

展示列表

```
' GET: Store
Function Index() As ActionResult
    Dim genres = New List(Of Genre)
    genres.Add(New Genre("法师"))
    genres.Add(New Genre("打野"))
    genres.Add(New Genre("ADC"))
    genres.Add(New Genre("上单"))
    Return View(genres)
```

End Function

1.3Model



```
@ModelType IEnumerable(Of MusicStore3.Genre)
@Code
    ViewData("Title") = "Index"
End Code
<h2>Browse Genres</h2>
<p>
    Select from @Model.Count() genres:
</p>
<ul>
    @For Each genre In Model
        @<li>@Html.ActionLink(genre.Name, "Browse", New With {Key .genre =
genre.Name})</li>
    Next
</ul>
```

1.1Controller



Controller向前台传Json

```
Function SaveMemo(ByVal title As String, ByVal thumb As String, ByVal detail As String) As JsonResult
    Dim context = New MemoDBContext()
    Dim m = New Memo()
    m.Title = title
    m.Thumb = thumb
    m.Detail = detail
    m.CreateDate = Date.Now().ToString()
    Dim id As Integer = context.InsertMemo(m)
    m.Id = id
    Return Json(m, JsonRequestBehavior.AllowGet)
End Function
```

1.2View



View和页面通过ViewBag传值
Controller :

```
Function Welcome(ByVal name As String, Optional ByVal times As Integer = 1) As ActionResult
    ViewBag.Message = "欢迎" & name
    ViewBag.times = times
    Return View()
End Function
```

Welcome.vbhtml

@Code

```
ViewData("Title") = "Welcome"
```

```
For i = 0 To ViewBag.times
```

```
@<h3> @ViewBag.Message @i.ToString </h3>
```

```
Next
```

End Code