گام به گام: ساختن یک پروژه MVC

مقدمه

این مطلب شیوه ساختن یک پروژه ASP.NET MVC در ویژوال استودیو را نشان می دهد. در این مطلب یک برنامه نمونه MVC نوشته و سپس آن را اجرا می کنید. بعد از آن برنامه را با اضافه کردن کنترلگر و نمایشگر سفارشی می کنید.

همچنین این مطلب نحوه استفاده از «توسعه آزمون محور» یا Test-Driven Development را نیز نشان می دهد. در این مطلب پروژه ای می سازید که شامل Unit test های برنامه MVC است.

پیش نیازها

برای تکمیل این مراحل به صورت گام به گام، به این ها نیاز دارید:

- Microsoft Visual Studio 2008 SP1
  - نکته: Visual Studio 2008 Standard Edition و Visual Web Developer Express ۲۰۰۸ از unit test پشتیبانی نمی کنند. می توانید از این نسخه ها برای ساختن و اجرای قسمت هایی از این «گام به گام» که مربوط به ساخت و اجرای برنامه های ASP.NET MVC است استفاده کنید. در صورت استفاده از این نسخه ها، نمی توانید از unit test که در این «گام به گام» اشاره شده است، استفاده کنید.
- ASP.NET MVC Beta. این نسخه را می توانید از صفحه ASP.NET MVC Microsoft ASP.NET در سایت ASP.NET دریافت کنید.

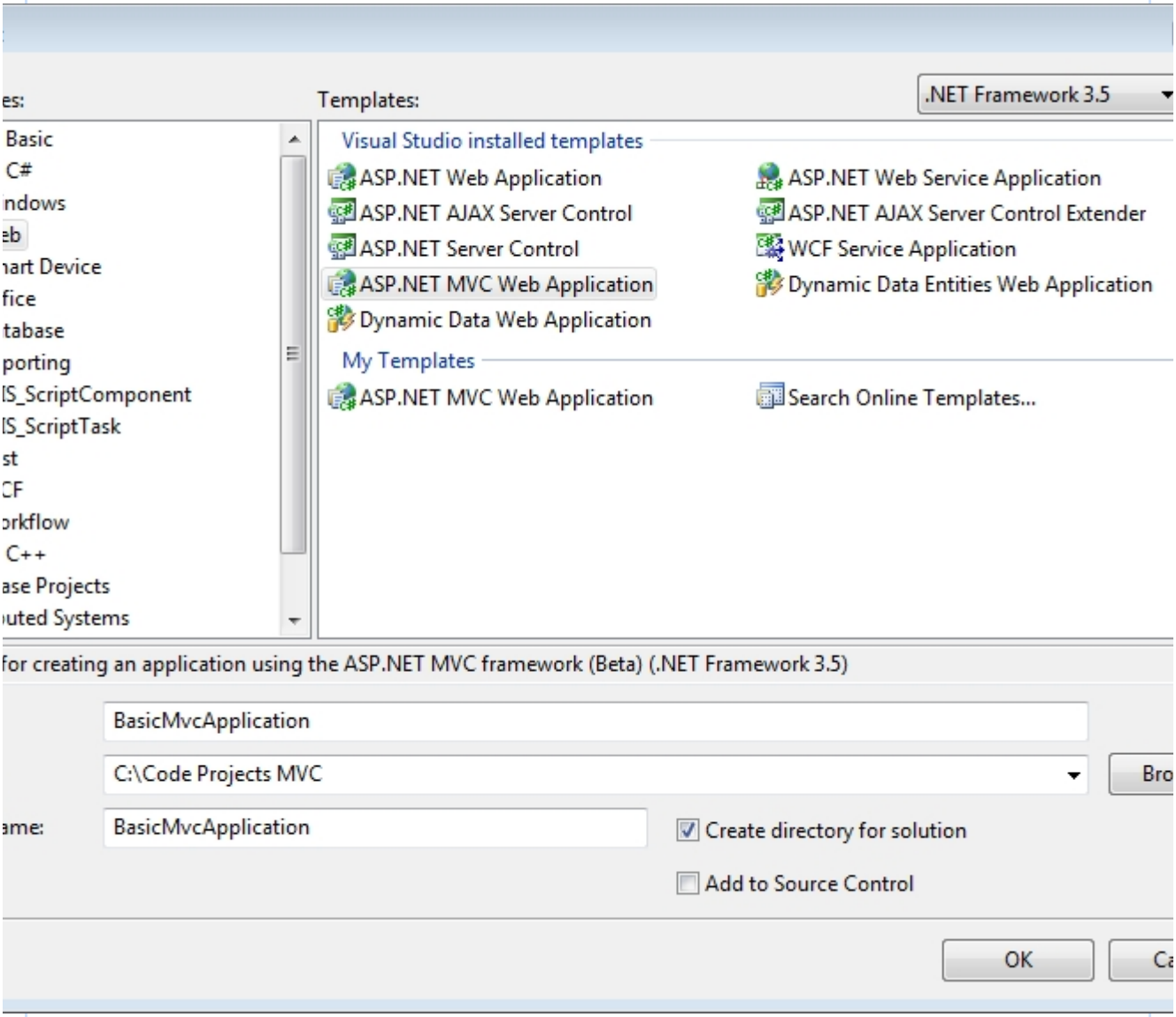
توضیح مترجم: در این قسمت مستندات رسمی هیچ اشاره ای به نصب ASP.NET MVC نشده است. با این حال قبل از ادامه اجرای مراحل این گام به گام، ابتدا ASP.NET MVC را نصب کنید و سپس ویژوال استودیو را re-start کنید.

ساخت یک پروژه MVC جدید

برای شروع، یک پروژه ASP.NET MVC Application بسازید. توجه کنید که پروژه Web Application می سازید نه Web Site.

گام به گام ساخت پروژه MVC

۱- در منوی **File** بر روی **Project New** کلیک کنید. کادر **New Project** همانند تصویر زیر نمایش داده می شود:



۲- به قسمت بالا سمت راست دقت کنید و مطمئن شوید که **.NET Framework 3.5** انتخاب شده است.

۳- در قسمت **Project Types** شاخه **Visual Basic** یا **C#** را باز کنید و سپس روی **Web** کلیک کنید.

٤- در قسمت **Visual Studio installed templates** گزینه **Application ASP.NET MVC Web** را انتخاب کنید.

٥- در قسمت **Name** عبارت **BasicMvcApplication** را وارد کنید.

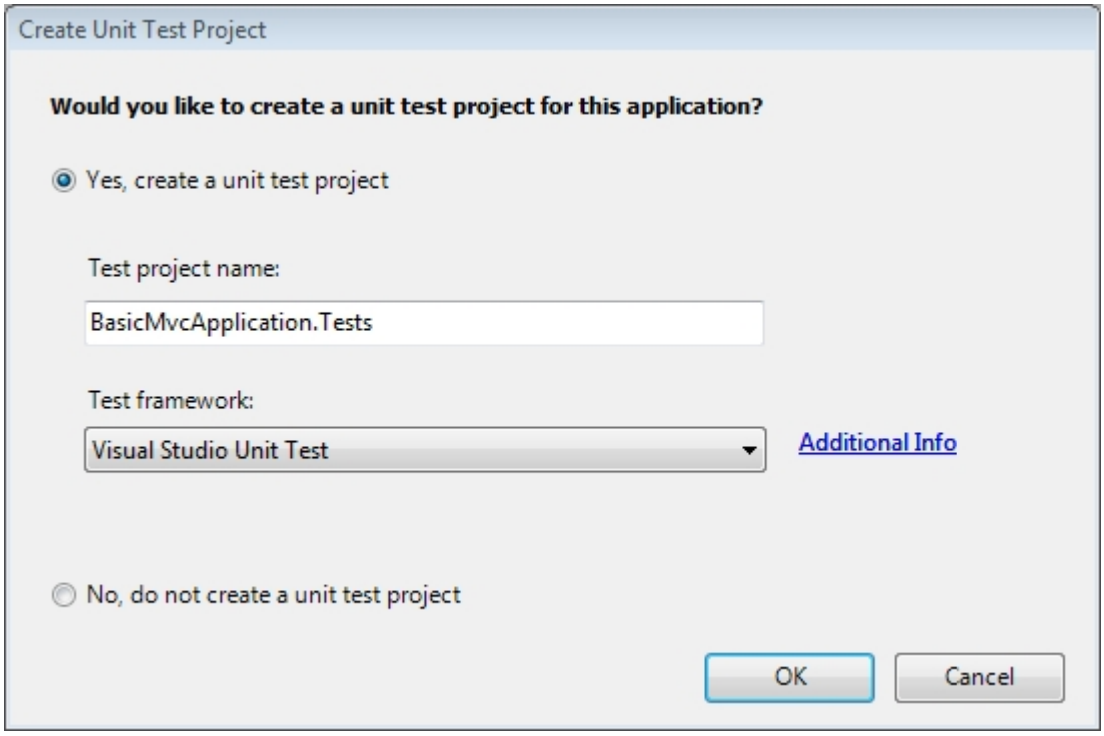
٦- در قسمت **Location** نام پوشه پروژه را مشخص کنید.

۷- اگر می خواهید نام Solution با نام پروژه متفاوت باشد، در قسمت **Solution Name** نام Solution خود را وارد کنید.

۸- گزینه **Create directory for solution** را انتخاب کنید.

۹- بر روی **OK** کلیک کنید.

در این مرحله پنجره **Create Unit Test Project** نمایش داده می شود.



نکته: اگر از Visual Studio 2008 Standard Edition یا Express Edition Visual Studio 2008 استفاده می کنید، پنجره **Create Unit Test Project** نمایش داده نمی شود و پروژه بدون پروژه تست ساخته می شود.

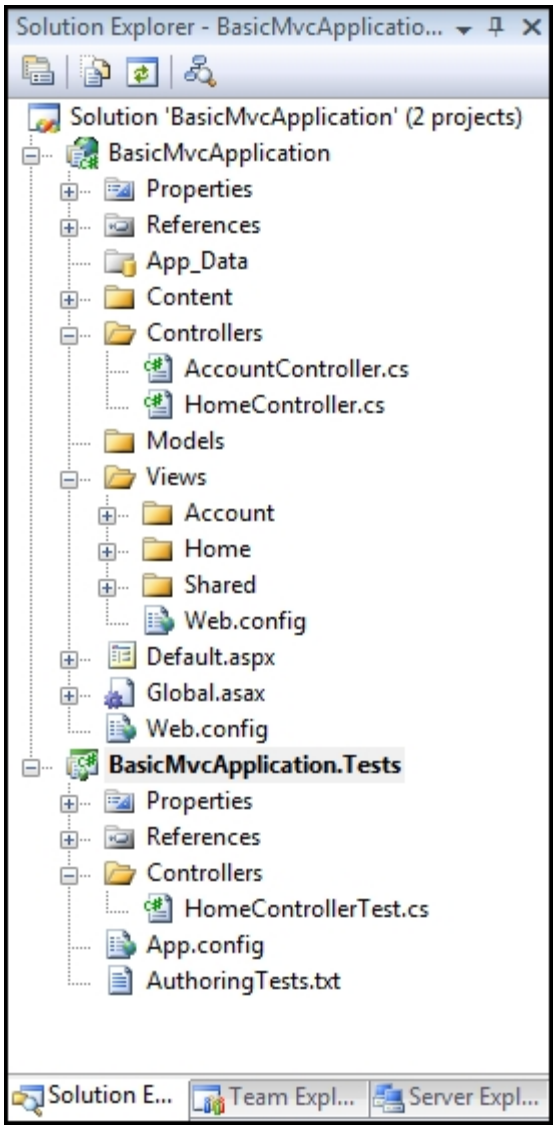
۱۰- گزینه **Yes, create a unit test project** را انتخاب کنید.

به صورت پیش فرض نام پروژه تست، همان نام پروژه است که Tests به انتهای آن اضافه شده است. اگر بخواهید می توانید این نام را تغییر دهید. همچنین پروژه تست به طور پیش فرض از Visual Studio Unit Test framework استفاده می کند. در صورتی که بخواهید نحوه استفاده از سایر ابزارهای آزمون را بیاموزید، صفحه Adding a Custom MVC Test Framework in Visual Studio را ببینید.

۱۱- بر روی **OK** کلیک کنید.

آشنایی با پروژه MVC

تصویر زیر ساختار پروژه ای که اکنون ایجاد شد را نشان می دهد:

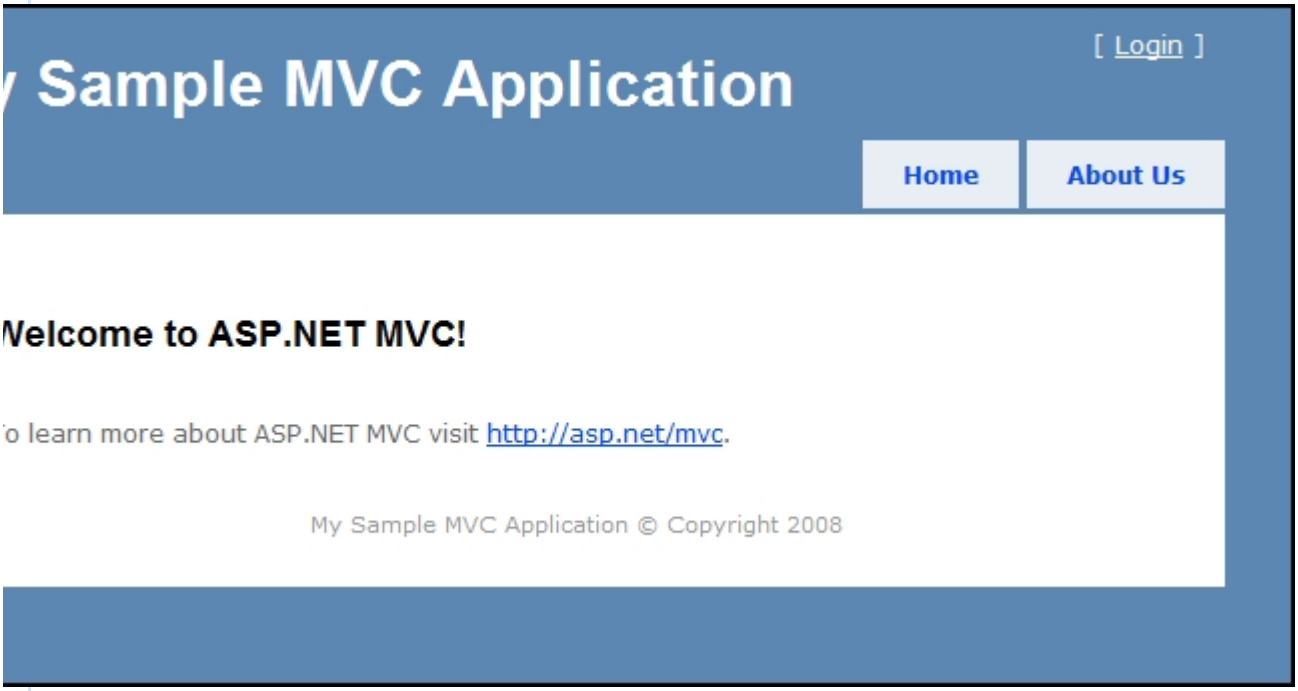


ساختار پروژه MVC با ساختار پروژه ASP.NET Web Site فرق می کند. پروژه MVC شامل پوشه های زیر است:

- پوشه **Content**: مخصوص پشتیبانی از فایل های محتوا. این پوشه شامل فایل CSS برنامه است. (از این پوشه برای نگهداری فایل های عکس، جاوا اسکریپت و ... استفاده می شود. -مترجم)
- پوشه **Controllers**: مخصوص کلاس های کنترلگر. این پوشه شامل کنترلگرهای نمونه برنامه به نام های AccountController و HomeController است. کلاس AccountContrller شامل عملیات لازم جهت ورود به سیستم (login) و خروج از سیستم (logout) و ثبت نام (register) است. HomeContrller شامل منطقی است که در زمان آغاز برنامه فراخوانی می شود. (مترجم: مشابه صفحه پیش فرض عمل می کند.)
- پوشه **Models**: مخصوص فایل های مدل داده (از قبیل فایل dbml. مرتبط با SQL LINQ to) و فایل های Data entity.
- پوشه **Views**: برای صفحات نمایشگر. این پوشه شامل سه پوشه داخلی است: Account و Home و Shared. پوشه Account شامل نمایشگرهایی است که به عنوان رابط کاربری (UI) ورود به سیستم (login) و خروج از آن (logout) و تغییر رمز عبور (password Change) و ثبت نام در سیستم (register) از آن ها استفاده می شود. پوشه Home شامل نمایشگر Index (صفحه آغازین پیش فرض برنامه) و

نمایشگر About است. پوشه Shared شامل نمایشگر master page برنامه است.

اگر از نسخه های ویژوال استودیو غیر از Edition Standard و Express Edition استفاده می کنید، یک پروژه آزمون (test) نیز ایجاد می شود. برنامه آزمون شامل پوشه ای به نام Controllers است که خود شامل کلاس HomeContrllerTest است. این کلاس برای هر متد عملیات در کلاس HomeContrller (یعنی Index و Home) یک آزمون unit test دارد. پروژه تازه ساخته شده برنامه کاملی است که بدون تغییر می توانید آن را کامپایل و اجرا کنید. تصویر زیر برنامه در حال اجرا را در مرورگر نشان می دهد:



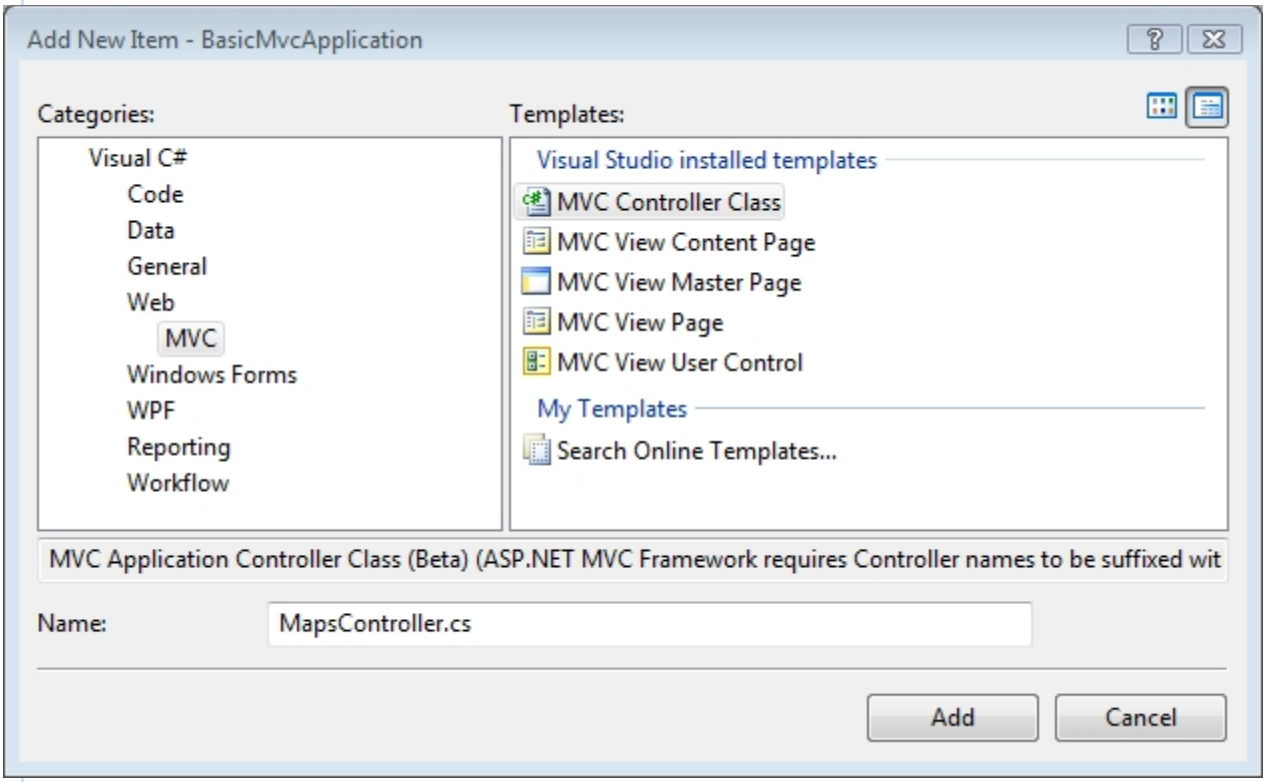
پروژه unit test نیز آماده کامپایل و اجرا است. در این گام به گام، یک کنترلگر به همراه یک متد عملیات و یک نمایشگر به برنامه اضافه می کنید و یک unit test برای متد جدید می نویسید.

اضافه کردن کنترلگر

اکنون کنترلگری به برنامه اضافه می کنید که شامل منطق دریافت نقشه شهرها از خدمات وب Microsoft Virtual Earth است.

برای اضافه کردن کنترلگر به برنامه MVC:

۱- در **Solution Explorer** روی پوشه **Controllers** راست کلیک کرده و **Add** و سپس **New Item** را انتخاب کنید. پنجره **Add New Item** نمایش داده می شود:



۲- در قسمت **Categories** شاخه **Visual Basic** یا **Visual C#** و سپس شاخه **Web** را باز کرده و بر روی **MVC** کلیک کنید.

۳- در قسمت **Templates** گزینه **MVC Controller Class** را انتخاب کنید.

٤- در قسمت **Name** عبارت **MapsContrller** را وارد کنید.

نکته: در ASP.NET MVC نام کنترلگرها حتماً باید به Contrller ختم شود مانند HomeController و GameController و MapsController.

٥- روی **Add** کلیک کنید.

ویژوال استودیو کلاس MapsController را به پروژه اضافه می کند.

ایجاد بدنه متد عملیات

برای اجرای تکنیک های توسعه آزمون محور (Development Test-Driven)، باید قبل از این که متدی را بنویسید، آزمون (تست) آن را بنویسید. با این حال برای این کار باید بدنه متد را آماده کنید. این متد در این گام به گام، ViewMaps است.

مراحل اضافه کردن بدنه متد عملیات:

۱- کلاس **MapsController** را باز کنید.

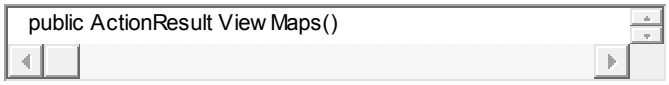
۲- متد **Index** را با متد **ViewMaps** که کد آن در ادامه می آید جایگزین کنید.

کد Visual Basic

```
view plaincopy to clipboardprint?
1.    Function ViewMaps()
2.        ' Add action logic here
3.        Throw New NotImplementedException()
4.    End Function
```

کد C#

```
view plaincopy to clipboardprint?
1.    public ActionResult ViewMaps()
2.    {
3.        // Add action logic here
4.        throw new NotImplementedException();
5.    }
```



اضافه کردن unit test برای متدهای عملیات

در این مرحله یک کلاس تست به پروژه BasicMvcApplication.Tests اضافه می کنید. در این کلاس یک unit test برای متد عملیات ViewMaps اضافه می کنید. این آزمون شکست می خورد زیرا بدنه متد عملیات ViewMaps یک استثناء پرتاب می کند. وقتی متد عملیات کاملاً پیاده سازی شد، آزمون موفق می شود.

مراحل اضافه کردن unit test به متدهای عملیات:

۱- در پروژه test بر روی پوشه **Controllers** کلیک راست نموده و گزینه **Add** و سپس گزینه **New Item** را انتخاب کنید.

پنجره **Add New Item** باز می شود.

۲- در قسمت **templates** گزینه **Class** را انتخاب کنید.

۳- در قسمت نام عبارت **MapsControllerTest** را وارد کنید.

٤- بر روی دکمه **Add** کلیک کنید.

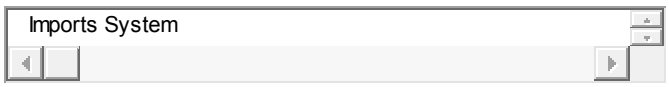
ویژوال استودیو کلاس **MapsControllerTest** را به پروژه آزمون (تست) اضافه می کند.

٥- کلاس **MapsControllerTest** را باز نموده و کد زیر را در آن وارد کنید. این کد unit test هایی برای متدهای عملیات -که بعد از این آن ها را تکمیل خواهید کرد- تعریف می کند.

کد ویژوال بیسیک:

```
view plaincopy to clipboardprint?
1.    Imports System
2.    Imports System.Collections.Generic
```

```
Imports System.Text
Imports System.Web.Mvc
Imports Microsoft.VisualStudio.TestTools.UnitTesting
Imports BasicMvcApplication

<TestClass()> Public Class MapsControllerTest


    Private testContextInstance As TestContext


    '''<summary>
    '''Gets or sets the test context which provides
    '''information about and functionality for the current test run.
    '''</summary>
    Public Property TestContext() As TestContext
        Get
            Return testContextInstance
        End Get
        Set(ByVal value As TestContext)
            testContextInstance = value
        End Set
    End Property


    <TestMethod()> Public Sub ViewMaps()
        ' Arrange
        Dim controller As MapsController = New MapsController()

        ' Act
        Dim result As ViewResult = controller.ViewMaps()

        ' Assert
        Dim viewData As ViewDataDictionary = result.ViewData
        Assert.AreEqual("My City Maps", viewData("Title"))
    End Sub
End Class
```



کد C#

view plaincopy to clipboardprint?

```
using System;
using System.Text;
using System.Collections.Generic;
using System.Linq;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System.Web.Mvc;
using BasicMvcApplication;
using BasicMvcApplication.Controllers;


namespace BasicMvcApplication.Tests.Controllers
{
    [TestClass]
    public class MapsControllerTest
    {
        [TestMethod]
        public void ViewMaps()
        {
            // Arrange
            MapsController controller = new MapsController();

            // Act
            ViewResult result = controller.ViewMaps() as ViewResult;

            // Assert
            ViewDataDictionary viewData = result.ViewData;
            Assert.AreEqual("My City Maps", viewData["Title"]);
        }
    }
}
```

```
using System;
```

۶- در **Solution Explorer** پروژه **BasicMvcApplication.Tests** را انتخاب نموده و سپس دکمه های Ctrl + F5 را بفشارید تا آزمون ها اجرا شوند.

آزمون با شکست مواجه می شود زیرا هنوز متد عملیات یک استثناء (Exception) پرتاب می کند.

اضافه کردن کد به متد عملیات

اکنون به متد عملیات **ViewMaps** از کلاس **MapsController** کدی برای تولید نمایشگر **Maps** اضافه می کنیم.

مراحل اضافه کردن کد به متد عملیات:

۱- کلاس **MapsController** را باز نموده و کد متد عملیات **ViewMaps** را با کد زیر جایگزین کنید:

کد ویژوال بیسیک:

view plaincopy to clipboardprint?

```
1.   Public Function ViewMaps()
2.       ViewData("Title") = "My City Maps"
3.       Return View()
4.   End Function
```

```
Public Function View Maps()
```

کد #C

view plaincopy to clipboardprint?

```
1.   public ActionResult ViewMaps()
2.   {
3.       ViewData["Title"] = "My City Maps";
4.       return View();
5.   }
```

```
public ActionResult View Maps()
```

۲- فایل را دخیره کنید.

اضافه کردن نمایشگر

در این مرحله، نمایشگر **Maps** را ایجاد می کنید. برای سازماندهی نمایشگرها ابتدا پوشه ای به نام **Maps** در پوشه **Views** ایجاد خواهید کرد.

مراحل اضافه کردن یک نمایشگر به برنامه MVC:

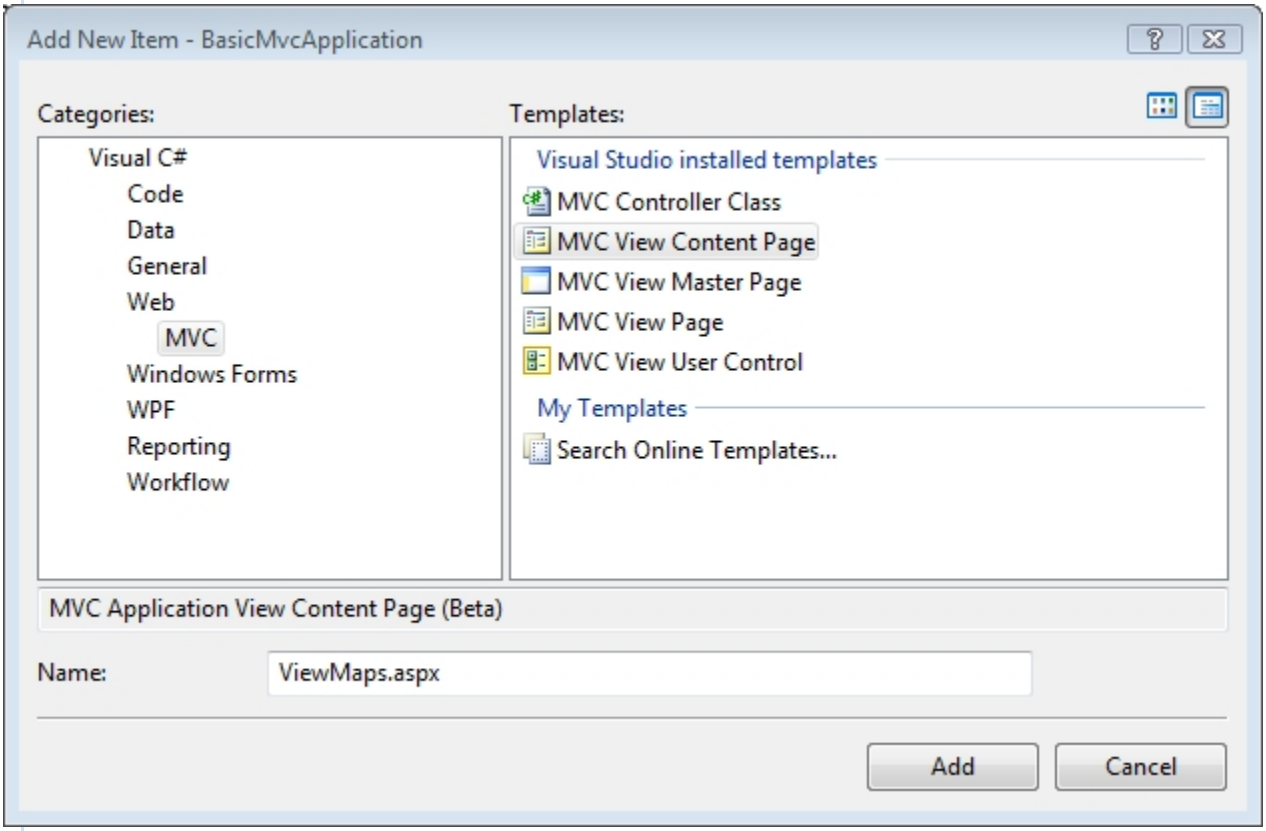
۱- در **Solution Explorer** بر روی پوشه **Views** کلیک راست نموده و **Add** و بعد از آن **New Folder** را انتخاب کنید.

۲- نام پوشه جدید را **Maps** بگذارید.

نکته: نمایشگرها در پوشه ای قرار می گیرند که نام آن از نام کلاس کنترلگر تبعیت می کند. برای مثال، نمایشگرهای مرتبط با کلاس کنترلگر **MapsController** در پوشه ای به نام **Maps** قرار می گیرند.

۳- بر روی پوشه **Maps** کلیک راست نموده، گزینه **Add** و سپس گزینه **Item New** را انتخاب کنید.

پنجره **Add New Item** نمایش داده می شود:



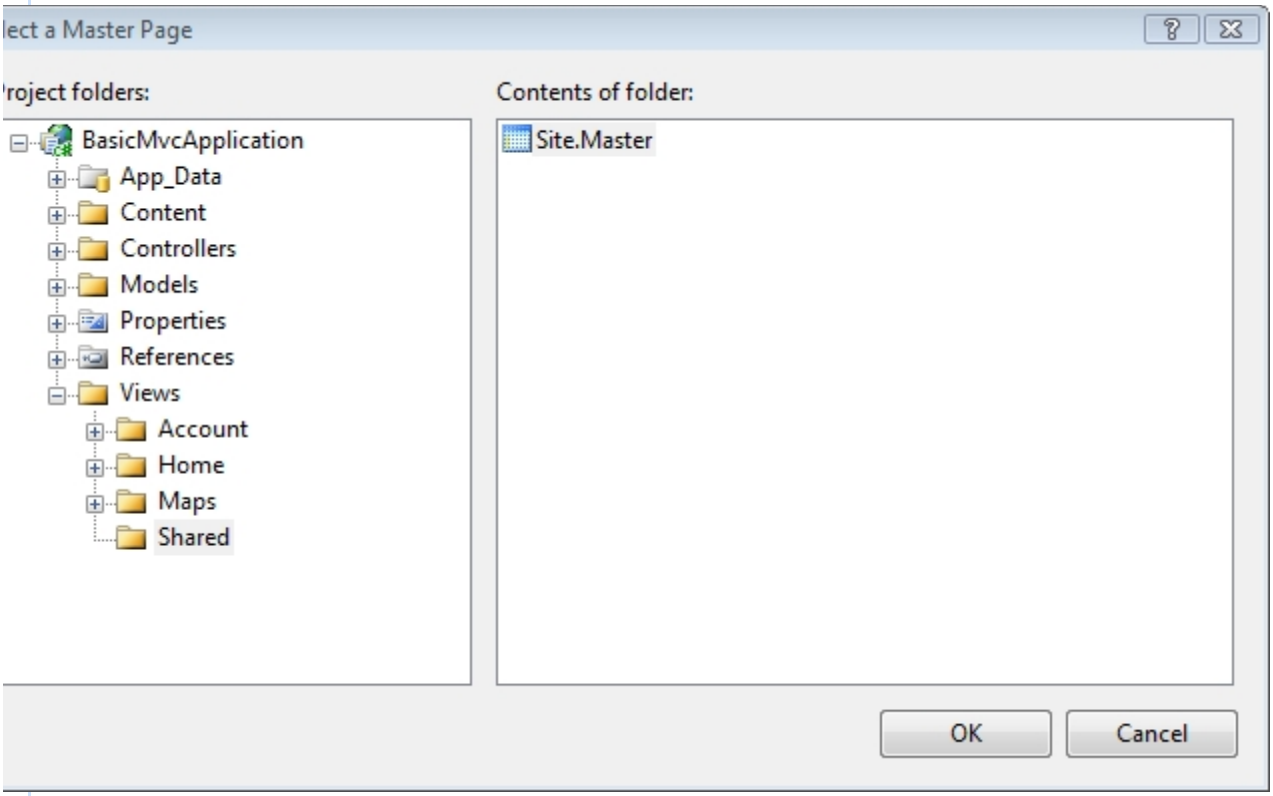
٤- در قسمت Categories شاخه Web را باز و سپس MVC را انتخاب کنید.

٥- در قسمت Templates گزینه **Page MVC View Content** را انتخاب کنید.

٦- در قسمت نام، عبارت ViewMaps.aspx را وارد کنید.

۷- دکمه Add را کلیک کنید.

پنجره **Select a Master Page** نمایش داده می شود:



۸- از قسمت Project Folders شاخه Views و سپس شاخه Shared را انتخاب کنید.

۹- در قسمت Contents of Folder فایل Site.Master را انتخاب کنید و سپس دکمه OK را کلیک کنید.

نمایشگر جدید در شاخه Maps اضافه می شود.

اضافه کردن کد به نمایشگر

در این مرحله، به نمایشگری که در مرحله قبل ساخته شد، کد اضافه می کنیم.

مراحل اضافه کردن کد به نمایشگر:

۱- نمایشگر ViewMaps.aspx را باز نموده و کد زیر را در قسمت Content آن وارد کنید:

view plaincopy to clipboardprint?

```
1.
2.   <h2>My City Maps</h2>
3.   Select map:
4.   <select onclick="GetMap(value);">
5.       <option value="Seattle">Seattle, WA</option>
6.       <option value="LasVegas">Las Vegas, NV</option>
7.       <option value="SaltLake">Salt Lake City, UT</option>
8.       <option value="Dallas">Dallas, TX</option>
9.       <option value="Chicago">Chicago, IL</option>
10.      <option value="NewYork">New York, NY</option>
11.      <option value="Rio">Rio de Janeiro, Brazil</option>
12.      <option value="Paris">Paris, France</option>
13.      <option value="Naples">Naples, Italy</option>
14.      <option value="Keta">Keta, Ghana</option>
15.      <option value="Beijing">Beijing, China</option>
16.      <option value="Sydney">Sydney, Australia</option>
```

```
</select>
<br />
<br />
<div id='earthMap' style="position:relative; width:400px; height:400px;">
</div>


<script charset="UTF-
8" type="text/javascript" src="http://dev.virtualearth.net/mapcontrol/mapcontrol.htm?v=6.2&mkt=en-us">
</script>
<script type="text/javascript">
    var map = null;
    var mapID = '';

    function GetMap(mapID)
    {
        switch (mapID)
        {
            case 'Seattle':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(47.6, -122.33), 10 ,'i', true);
                break;
            case 'LasVegas':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(36.17, -115.14), 10 ,'i' ,true);
                break;
            case 'SaltLake':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(40.75, -111.89), 10 ,'i' ,true);
                break;
            case 'Dallas':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(32.78, -96.8), 10 ,'i' ,true);
                break;
            case 'Chicago':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(41.88, -87.62), 10 ,'i' ,true);
                break;
            case 'NewYork':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(40.7, -74), 10 ,'i' ,true);
                break;
            case 'Rio':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(-22.91, -43.18), 10 ,'i' ,true);
                break;
            case 'Paris':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(48.87, 2.33), 10 ,'i' ,true);
                break;
            case 'Naples':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(40.83, 14.25), 10 ,'i' ,true);
                break;
            case 'Keta':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(5.92, 0.983), 10 ,'i' ,true);
                break;
            case 'Beijing':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(39.91, 116.39), 10 ,'i' ,true);
                break;
            case 'Sydney':
                map = new VEMap('earthMap');
                map.LoadMap(new VELatLong(-33.86, 151.21), 10 ,'i' ,true);
          }
    }
</script>
```

```
<h2>My City Maps</h2>
```

این قطعه کد یک لیست کشویی (Drop-Down List) برای انتخاب نقشه تعریف می کند و جاوا اسکریپت برای دریافت اطلاعات نقشه از خدمات وب Virtual Earth مایکروسافت است.

۲- فایل را ذخیره کنید.

اضافه کردن یک قفسه (Tab) به منوی صفحه Master

حالا به منوی صفحه master آیتمی اضافه می کنیم که متد عملیات ViewMaps را فراخوانی می کند.

مراحل اضافه کردن یک قفسه به صفحه Master:

۱- فایل Site.master را از پوشه Views/Shared باز کنید و در div با شناسه menucontainer لیست غیر مرتب (ul) را پیدا کنید.

۲- خط زیر را بین خطوط مربوط به قفسه های Index و About Us اضافه کنید.

```
view plaincopy to clipboardprint?
1.    <li><%= Html.ActionLink("Maps", "ViewMaps", "Maps")%></li>
```

<%= Html.ActionLink("Maps", "View Maps", "Maps")%>

نکته: متد ActionLink یک متد کمکی (Helper Method) است که پیوندی به یک متد عملیات ایجاد می کند. این متد سه پارامتر می گیرد: متن پیوند، نام متد عملیات و نام کنترلگر.

۳- فایل را ذخیره کنید.

آزمون و اجرای برنامه MVC

حالا می توانید برنامه را تست کنید.

مراحل آزمون برنامه MVC:

۱- از منوی Test بر روی Run کلیک نموده و سپس برای اجرای Unit test ها گزینه All Tests in Solution را انتخاب کنید.

نتایج آزمون ها در پنجره Test Results نمایش داده می شود. این بار تست ها با موفقیت اجرا می شوند.

۲- در Solution Explorer پروژه BasicMvcApplication را انتخاب نموده و برای اجرای برنامه کلیدهای Ctrl + F5 را بزنید.

صفحه Index.aspx که شامل قفسه های صفحه master است، نمایش داده می شود.

۳- بر روی قفسه Maps کلیک کنید.

صفحه Maps Page نمایش داده می شود. شهری را انتخاب کنید تا نقشه آن به شما نمایش داده شود.

y Sample MVC Application

[Login]

Home | Maps | About Us

Maps Page

Select map: Seattle, WA

My Sample MVC Application © Copyright 2008

مراحل بعدی

این گام به گام یک نگاه اولیه به ASP.NET MVC Framework و همچنین توسعه آزمون محور در برنامه های MVC به شما می دهد.