

## آموزش MVC

### Model - View - Controller

MVC3 آخرین تکنولوژی میکروسافت در زمینه تولید صفحات وب (تا این لحظه ، یعنی مارچ ۲۰۱۱) می باشد که کم کم جایگزین روش قبلی تولید صفحات وب (Web Forms) می شود و جالب اینجاست که به شدت از سوی میکروسافت تبلیغ می شود و منابع آن به سرعت انتشار می یابند. این تکنولوژی که در حال حاضر نسخه ۳ آن منتشر شده است دارای مزیت‌های زیر نسبت به تکنولوژی قبلی (Web Forms) می باشد(در واقع موارد ذیل مشکلات وب فرمها هستند).

- در وب فرمها ، برنامه نویس کنترل زیادی روی خروجی HTML تولید شده توسط ASP.NET ندارد و اگر شما سورس کد صفحه را ببینید ، شاهد کد HTML شلوغی هستید که مشکل می توانید از آن سر در بیاورید
- در وب فرمها با رد و بدل شدن ViewState در هنگام ارسال به سرور ، حجم آن زیاد می شود و صفحه به اصطلاح پف می کند
- ایجاد واحد تست (Unit Test) در وب فرمها مشکل است
- چرخه رد و بدل شدن فایل بین کلاینت و سرور پیچیده است

تکنولوژی MVC تمام مشکلات فوق را حل می کند و مزایایی نیز دارد از جمله :

- Method Request است یعنی شما می توانید به جای فراخوانی کل یک صفحه ، یک متد را فراخوانی کنید
- Partly Open Source است . یعنی بخشی از کدهای اصلی آن "باز" است و می توانید تحت استاندارد نرم افزارهای متن باز ، آن را ویرایش کنید

من در ابتدا تصمیم داشتم که در ۵ الی ۶ مقاله کل مبحث را خلاصه کنم ولی ترجیح دادم تا به جای آن ، بصورت گام به گام یک پروژه MVC را شروع کنم و در طی آن مطالب جدید را بیان کنم. اما قبل از شروع :

#### -نصب برنامه:

MVC2 در ویژال استودیو ۲۰۱۰ نگارش Ultimate از پیش نصب شده است ولی اگر بدنبال نصب MVC3 هستید ، از آدرس <http://www.microsoft.com/web> ابزار WebMatrix را نصب کنید. این ابزار پس از نصب ، پنجره 3 Web Platform Installer را در

اختیار شما می گذارد که نه تنها می توانید MVC3 را نصب کنید ، بلکه می توانید برنامه های مفید دیگر ، مثل تنظیمات توصیه شده مایکروسافت برای IIS7 یا موتور (engine) بسیاری از CMS های معروف مثل DotNetNuke ، Joomla ، Umbraco ، Drupal را نیز نصب کنید (یکی به عجایب هفت گانه دنیا اضافه شده !!!)

### -مبانی:MVC

من چون نمی خواهم خیلی وارد مباحث تئوریک شوم بصورت خلاصه توضیح می دهم و زمانی که پروژه را شروع کردیم ، خودتان این مفاهیم را بهتر متوجه می شوید . برای شروع از MVC2 آغاز می کنیم و پس از طرح مباحث اصلی MVC3 را نیز شرح می دهم .

در MVC سه جزء اصلی وجود دارد :

- Model
- Controller
- View

Model کلاسهای برای ارتباط با دیتابیس و استفاده از تکنیک هایی مثل Linq و Entity Framework هستند

Controller قلب MVC است و کنترل کننده درخواستهای کاربر است ، کلاسهای هستند که در آنها ، متدهای (اکشن های) مختلف جهت پاسخگویی به درخواست کاربر و انتخاب Model و یا View خروجی مناسب ، وجود دارد.

View رابط کاربری است یعنی آنچه کاربر باید ببیند را در آن نمایش می دهیم.

**-ایجاد پروژه جدید :** برای ایجاد پروژه جدید مطابق معمول از منوی File ویزال استودیو ، گزینه New و سپس Project را

انتخاب کنید و سپس یکی از قالبهای زیر را انتخاب کنید:

- ASP.NET MVC2 Web Application
- ASP.NET MVC2 Empty Web Application

(البته اگر MVC3 را نصب کرده باشید ، MVC3 را هم می بینید) تفاوت دو قالب فوق در این است که اولی شامل یک سری الگوهای آماده ، مثل Membership دات نت فریم ورک ، مراجع JQuery ، کلاسهای CSS از پیش ساخته شده است ولی دومی (Empty) اینها را ندارد و خودتان باید آنها را اضافه کنید برای شروع ، قالب اول را انتخاب کنید، نام و مسیر مناسبی برای آن

انتخاب کنید و پروژه را ایجاد نمایید ، از شما سوال می شود که آیا تمایلی به ایجاد واحد تست دارید یا خیر (که من برای شروع خیر را انتخاب می کنم، توی شکل زیر Yes انتخاب شده : !)

Would you like to create a unit test project for this application?

Yes, create a unit test project

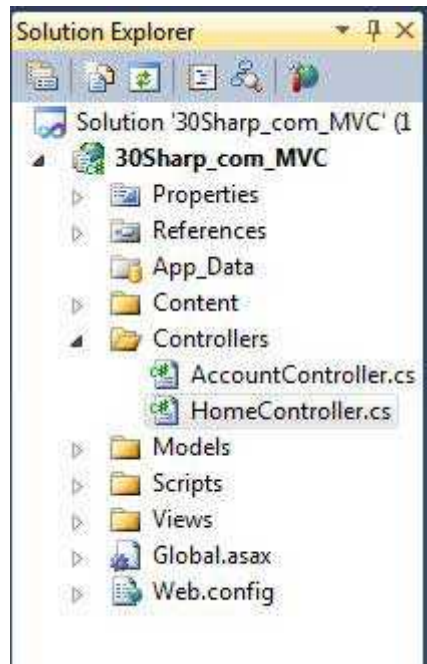
Test project name:  
30Sharp\_com\_MVC.Tests

Test framework:  
Visual Studio Unit Test [Additional Info](#)

No, do not create a unit test project

OK Cancel

پس از ایجاد باید شکل Solution Explorer شما به صورت زیر باشد:



همانطور که می بینید پوشه های مجزایی برای هر دسته از فایلها وجود دارد : مثل Model , View , Controller پوشه Script شامل فایلهای جاوا اسکریپت و JQuery و MicrosoftAjax است . پوشه Content هم محلی برای نگهداری فایلهای CSS و تصاویر است. اگر برنامه را اجرا کنید (F5) باید صفحه ای آبی رنگ با پیام Welcome to ASP.NET MVC! مشاهده کنید .

برای شروع کار فایل HomeController.cs از فولدر Controllers را باز کنید . کدهای زیر را در آن می بینید :

```
namespace _30Sharp_com_MVC.Controllers
{
    public class HomeController : Controller
    {
        [HandleError]
        public class HomeController : Controller
        {
            public ActionResult Index()
            {
                ViewData["Message"] = "Welcome to ASP.NET MVC!";
            }
        }
    }
}
```

```
        return View();
    }

    public ActionResult About()
    {
        return View();
    }
}
}
```

نکاتی که از کد فوق برداشت می کنیم :

هر فایل از نوع Controller با کلمه دلخواهی شروع می شود و بدنبال آن کلمه Controller می آید و این نام دلخواه در URL سایت تاثیر می گذارد . قالب URL در MVC به فرم زیر است :

```
Web site address/Controller/Action
```

اصلی سایت اکشن / کنترلر / آدرس

برای مثال برنامه را اجرا کنید ، روی لینک About کلیک کنید و URL صفحه را ببینید ، چیزی شبیه این می بینید (به جز شماره پورت که متفاوت است )

```
http://localhost:34256/Home/About
```

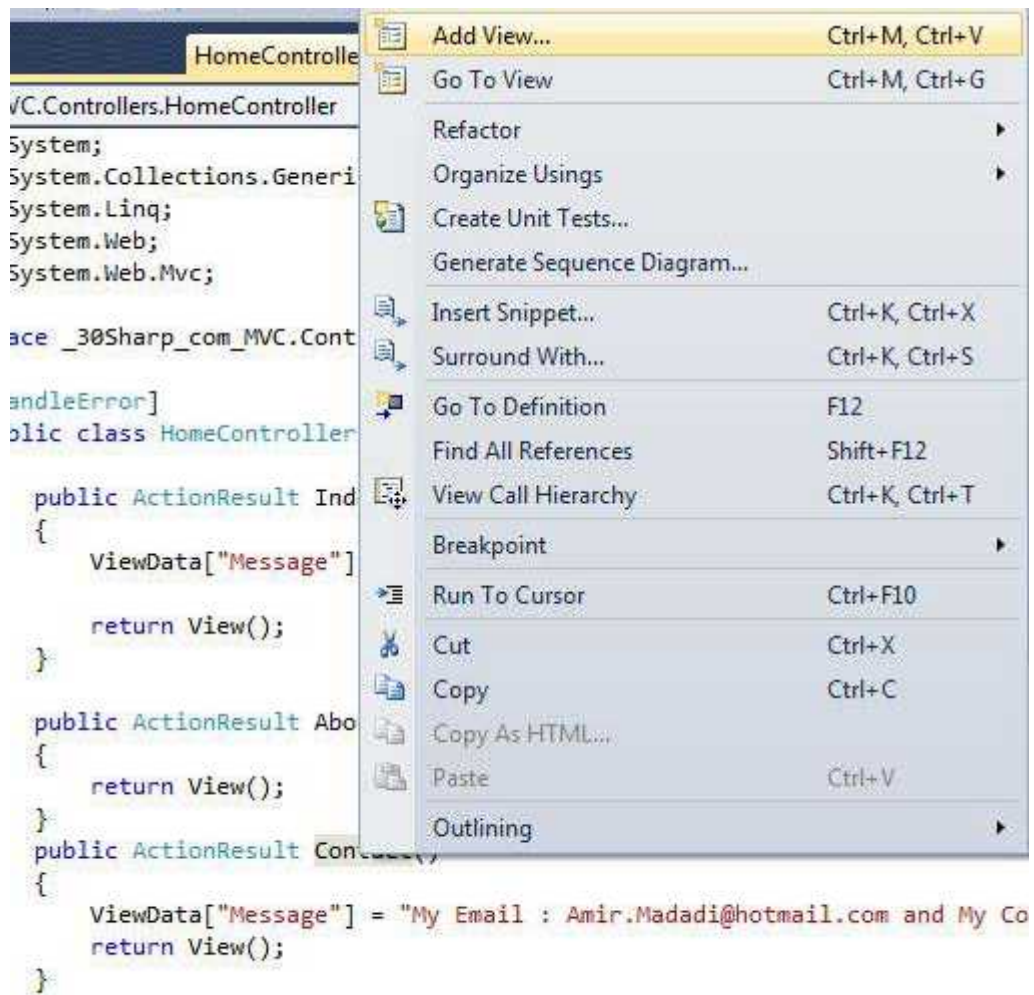
به متدها و رخدادهای صفحه در ام وی سی ، Action می گویند و در هر کنترلر می توانید چندین اکشن بیاورید و یا Overload کنید خروجی اکشن ها ، به صورت پیش فرض ActionResult است ولی می تواند هر نوع دیگری باشد (مثلا . string) در یک کنترلر شما ممکن است چندین اکشن داشت که برای هر اکشن یک View ایجاد می کنید و در آن ، مشخص می کنید که کاربر

چه خروجی را باید مشاهده کند . برای مثال زیر اکشن About ، من اکشن دیگری به نام Contact ایجاد می کنم و در آن اطلاعات خود را در Message ذخیره می کنم :

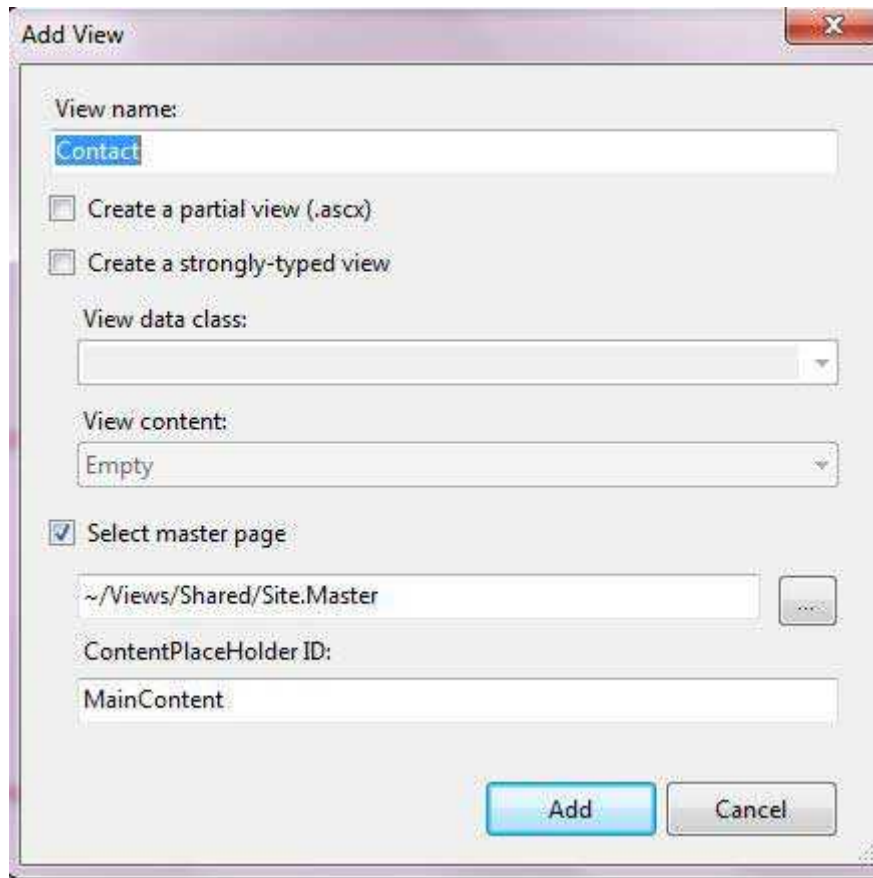
```
public ActionResult Contact()
{
    ViewData["Message"] = "My Email : Amir.Madadi@hotmail.com and My Contact
is 0422760153";
    return View();
}
```

توضیح اینکه شئی ViewData از نوع دیکشنری است که در آن می توانید کلید تعریف کنید و به آن مقدار دهید که من در اینجا کلیدی به نام Message تعریف کرده ام و اطلاعات تماس را درون آن ریخته ام .

حال باید برای این اکشن یک View ساخت تا مشخص کنیم کاربر چه چیزی را ببیند. برای این کار روی اکشن کلیک راست می کنیم و گزینه Add View را انتخاب می کنیم :



سپس پنجره زیر ظاهر می شود:



در مورد امکانات این پنجره بعدا صحبت می کنیم ، فعلا پیش فرضها را قبول کنید و روی Add کلیک کنید . خواهید دید که فایلی به نام Contact.aspx در فولدر View ایجاد می شود . در تگ `Asp:Contect id="Content2"` خط زیر را اضافه کنید :

```
<h2><%: ViewData["Message"] %></h2>
```

کد صفحه Contact.aspx شما باید شبیه این باشد :

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<dynamic>" %>
<asp:Content ID="Content1" ContentPlaceHolderID="TitleContent" runat="server">
    Contact
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Contact</h2>
    <h2><%: ViewData["Message"] %></h2>
</asp:Content>
```



حال برنامه را اجرا کنید و بعد از شماره پورت تایپ کنید /Home/Contact/ یعنی چیزی شبیه این :

```
http://localhost:34256/home/contact
```

اگر متن تایپ شده را در صفحه می بینید، به خود تبریک بگویید ، چون اولین صفحه وب خود را با تکنولوژی MVC ساخته اید .

قبل از پایان ، نگاهی به ساب فولدرهای View بیندازید ، خواهید دید که به ازای هر Controller یک فولدر وجود دارد که در آن View های مربوطه نگهداری می شوند (در حال حاضر باید Account ، Home و Shared را ببینید) که فولدر Shared برای ذخیره کردن اجزای مشترک مثل Master Page ها استفاده می شود .

خب تا اینجا کار یک پروژه جدید MVC ایجاد کردیم ، کمی در مورد ساختار MVC صحبت کردیم و یک View جدید به پروژه اضافه کردیم . در مقاله بعدی صفحاتی با کارایی بیشتر ایجاد می کنیم، تا آن موقع مروری بر دانسته های خود درباره Linq to SQL داشته باشید ، چرا که بزودی از آن در صفحات Model استفاده می کنیم