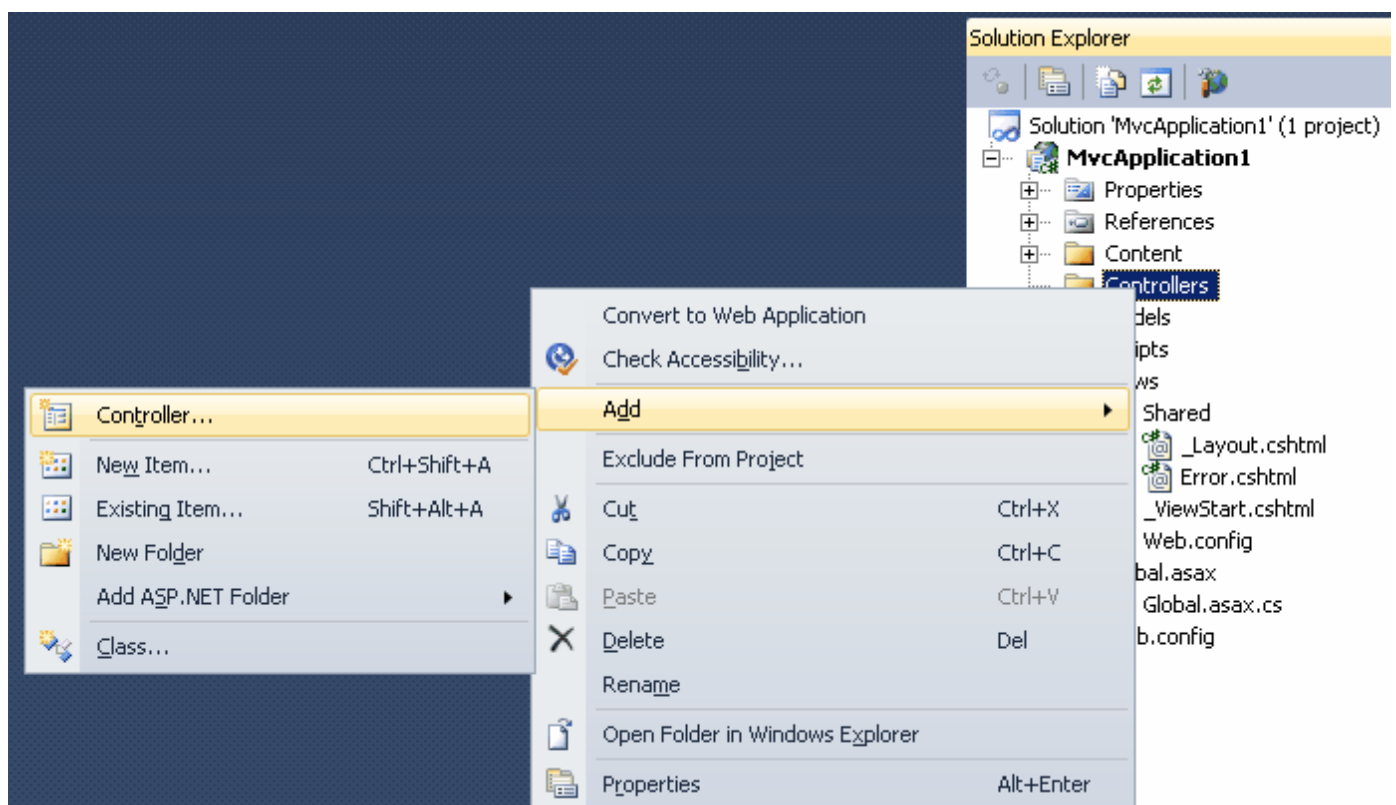


## بررسی نحوه ارتباطات بین اجزای مختلف الگوی MVC در ASP.NET MVC

اینبار برخلاف قسمت قبل، قالب پروژه خالی ASP.NET MVC را در VS.NET انتخاب کرده (ASP.NET MVC 3 Web Application) و بعد انتخاب قالب Empty نمایش داده شده) و سپس پروژه جدیدی را شروع می‌کنیم. View Engine را هم Razor در نظر خواهیم گرفت. پس از ایجاد ساختار اولیه پروژه، بدون اعمال هیچ تغییری، برنامه را اجرا کنید. بلافاصله با پیغام The resource cannot be found یا 404 یافت نشد، مواجه خواهیم شد. همانطور که در پایان قسمت دوم نیز ذکر شد، پردازش‌ها در ASP.NET MVC از کنترلرها شروع می‌شوند و نه از صفحات وب. بنابراین برای رفع این مشکل نیاز است تا یک کلاس کنترلر جدید را اضافه کنیم. به همین جهت بر روی پوشه استاندارد Controllers کلیک راست کرده، از منوی ظاهر شده قسمت Add، گزینه‌ی Controller را انتخاب کنید:



در صفحه بعدی که ظاهر می‌شود، نام HomeController را وارد کنید (با توجه به اینکه مطابق قراردادهای ASP.NET MVC، نام کنترلر باید به کلمه Controller ختم شود). البته لازم به ذکر است که این مراحل را به همان شکل متداول مراجعه به منوی Project و انتخاب Add Class و سپس ارث بری از کلاس Controller نیز می‌توان انجام داد و طی این مراحل الزامی نیست. کلاسی که به صورت خودکار از طریق منوی Add Controller یاد شده ایجاد می‌شود، به شکل زیر است:

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcApplication1.Controllers
{
    public class HomeController : Controller
    {
        //
        // GET: /Home/
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

### سؤال:

در قسمت دوم عنوان شد که کنترلر باید کلاسی باشد که اینترفیس `IController` را پیاده سازی کرده است، اما در اینجا ارث بری از کلاس `Controller` را شاهد هستیم. جریان چیست؟! سلسله مراتبی که بکار گرفته شده به صورت زیر است:

```
public abstract class ControllerBase : IController
public abstract class Controller : ControllerBase
```

`ControllerBase`، اینترفیس `IController` را پیاده سازی کرده و سپس کلاس `Controller` از کلاس `ControllerBase` مشتق شده است. شاید بپرسید که این همه پیچ و تاب برای چیست؟! مشکلی که با اینترفیس خالص وجود دارد، عدم نگارش پذیری آن است. به این معنا که اگر متدی یا خاصیتی در نگارش بعدی به این اینترفیس اضافه شد، هیچکدام از پروژه‌های قدیمی دیگر کامپایل نخواهند شد و باید ابتدا این متد یا خاصیت جدید را نیز لحاظ کنند. اینجا است که کار کلاس‌های `abstract` شروع می‌شود. در یک کلاس `abstract` می‌توان پیاده سازی پیش فرضی را نیز ارائه داد. به این ترتیب مصرف کننده نهایی کلاس `Controller` متوجه این تغییرات نخواهد شد. اگر برنامه نویس «من» باشم، شما رو وادار خواهم کرد که متدهای جدید اینترفیس تعریفی‌ام را پیاده سازی کنید! همینه که هست! اما اگر طراح مایکروسافت باشد، بلافاصله انبوهی از جماعت ایرادگیر که بالای 100 تا از کنترلرهای اون‌ها الان فقط در یک پروژه از کار افتاده، ممکن است جلوی دفتر مایکروسافت دست به خود سوزی بزنند! اینجا است که مایکروسافت مجبور است تا این پیچ و تاب‌ها را اعمال کند که اگر روزی متدی در اینترفیس `IController` بنابر نیازهای جدید در نظر گرفته شد، بتوان سریع یک پیاده سازی پیش فرض از آن‌را در کلاس‌های `abstract` یاد شده قرار داد (یکی از تفاوت‌های مهم کلاس‌های `abstract` با اینترفیس‌ها) تا جماعت ایرادگیر و نقزن متوجه تغییری نشوند و باز هم پروژه‌های قدیمی بدون مشکل کامپایل شوند. تابلال به فلسفه وجودی کلاس‌های `abstract` از این دیدگاه فکر کرده بودید؟!

بعد از این توضیحات و کارها، اگر اینبار برنامه را اجرا کنیم، خطای زیر نمایش داده می‌شود:

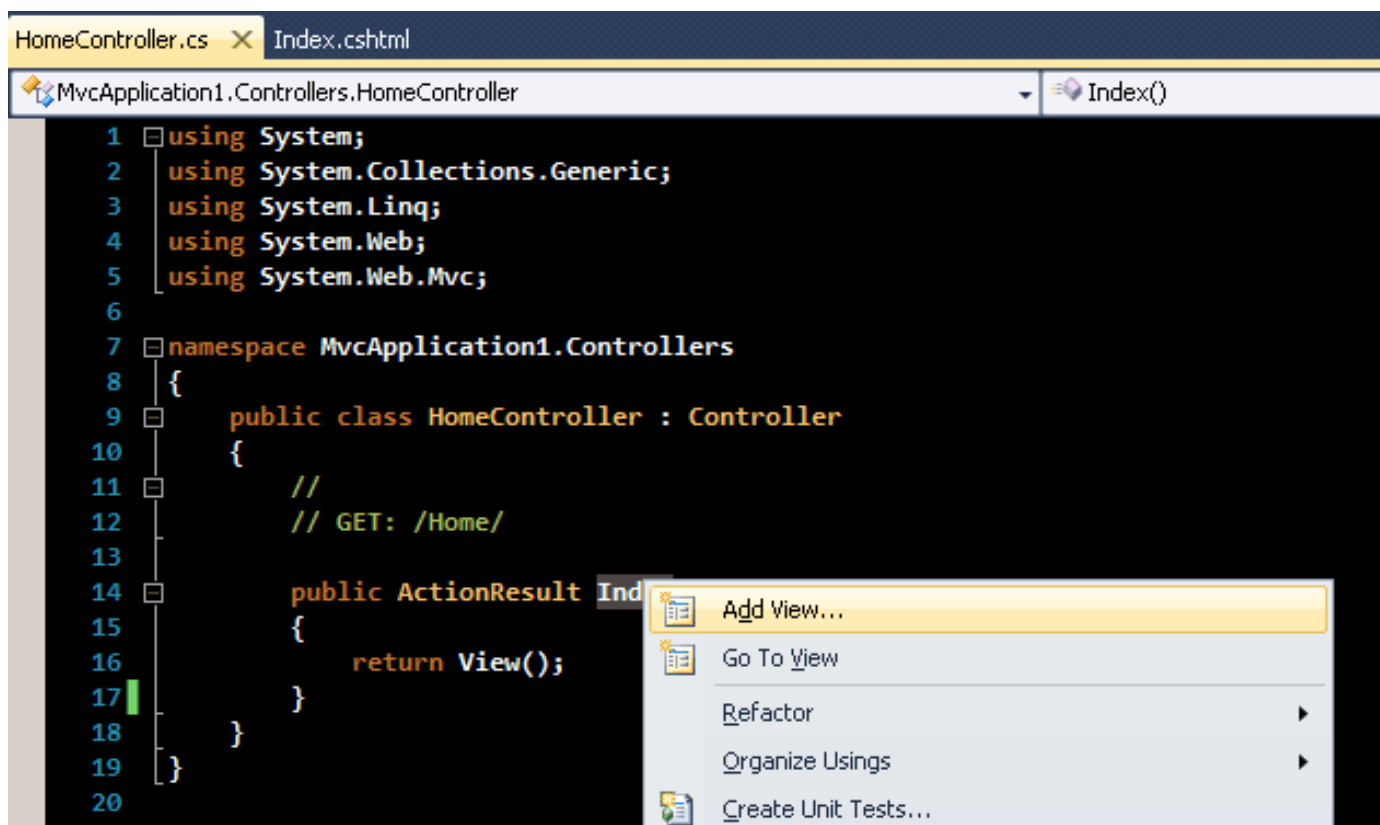
```
The view 'Index' or its master was not found or no view engine supports the searched locations. The
following locations were searched:
~/Views/Home/Index.aspx
~/Views/Home/Index.ascx
~/Views/Shared/Index.aspx
~/Views/Shared/Index.ascx
~/Views/Home/Index.cshtml
~/Views/Home/Index.vbhtml
~/Views/Shared/Index.cshtml
~/Views/Shared/Index.vbhtml
```

همانطور که ملاحظه می‌کنید چون نام کنترلر ما `Home` است، فریم ورک در پوشه استاندارد `Views` در زیر پوشه‌ای به همان نام

Home، به دنبال یک سری فایل می‌گردد. فایل‌های aspx مربوط به View Engine ایی به همین نام بوده و فایل‌های vbhtml و cshtml مربوط به View Engine دیگری به نام Razor هستند.

بنابراین نیاز است تا یکی از این فایل‌ها را در مکان‌های یاد شده ایجاد کنیم. برای این منظور حداقل دو راه وجود دارد. یا دستی اینکار را انجام دهیم یا اینکه از ابزار توکار خود VS.NET برای ایجاد یک View جدید استفاده کنیم.

برای ایجاد View ایی مرتبط با متد Index (در ASP.NET MVC نام دیگر متدهای قرار گرفته در یک کنترلر، Action نیز می‌باشد)، روی خود متد کلیک راست کرده و گزینه Add View را انتخاب کنید:



در صفحه بعدی ظاهر شده، پیش فرض‌ها را پذیرفته و بر روی دکمه Add کلیک نمائید. اتفاقی که رخ خواهد داد شامل ایجاد فایل Index.cshtml، با محتوای زیر است (با توجه به اینکه زبان پروژه سی شارپ است و View Engine انتخابی Razor می‌باشد، cshtml تولید گردید و گر نه vbhtml ایجاد می‌شد):

```
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
```

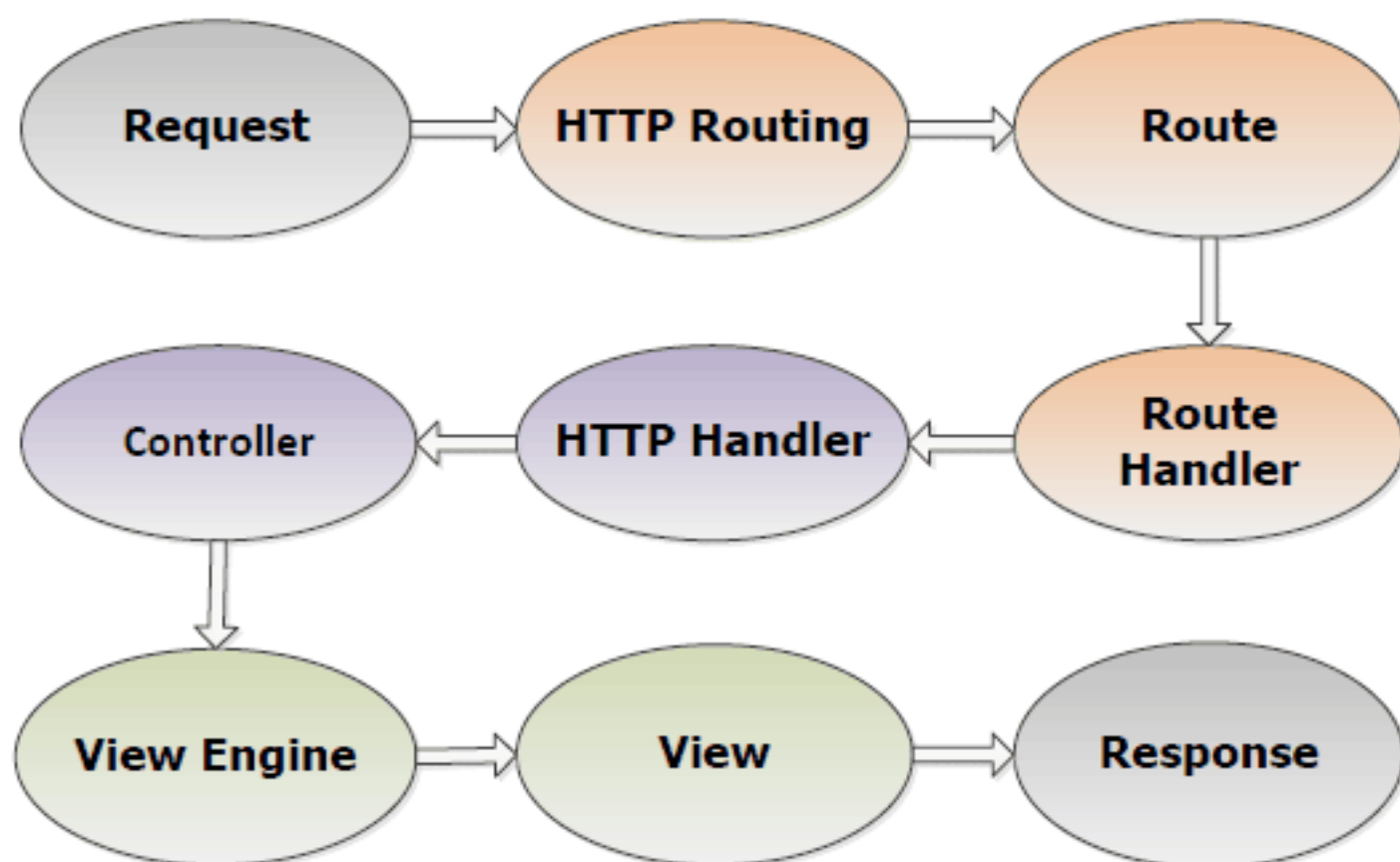
مجددا برنامه را اجرا کنید. اینبار بدون خطایی کلمه Index را در صفحه تولیدی می‌توان مشاهده کرد. نکته جالب این فایل‌های View جدید، عدم مشاهده ویژگی‌های runat=server و سایر موارد مشابه است.

#### چند سؤال مهم:

در حین ایجاد اولین کنترلر جهت نمایش صفحه پیش فرض برنامه، نام HomeController انتخاب شد. چرا مثلا نام TestController وارد نشد؟ برنامه از کجا متوجه شد که باید حتما این کنترلر را پردازش کند. نقش متد Index چیست؟ آیا حتما

باید Index باشد و در اینجا نام دیگری را نمی‌توان وارد کرد؟ «قرارداد» پردازشی این‌ها کجا تنظیم می‌شود؟ فریم ورک، این سیم کشی‌ها را چگونه انجام می‌دهد؟ پاسخ به تمام این سؤال‌ها، در ویژگی «مسیریابی یا Routing» نهفته است. فایل Global.asax.cs برنامه را باز کنید. تعاریف مرتبط با مسیریابی پیش فرض را در متد RegisterRoutes آن می‌توان مشاهده کرد:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter defaults
    );
}
```



پروسه هدایت یک درخواست HTTP به یک کنترلر، در اینجا مسیریابی یا Routing نامیده می‌شود. این قابلیت در فضای نام System.Web.Routing تعریف شده است و باید دقت داشت که جزو ASP.NET MVC نیست. این امکانات جزو ASP.NET Runtime است و به همراه دات نت 3.5 سرویس پک یک برای اولین بار ارائه شد. بنابراین جهت استفاده در ASP.NET Web forms نیز مهیا است. در ASP.NET MVC از این امکانات برای ارسال درخواست‌ها به کنترلرها استفاده می‌شود. در متد routes.MapRoute فراخوانی شده‌ای که در کدهای بالا ملاحظه می‌کنید، کار نگاشت یک URL به Action‌های یک کنترلر صورت می‌گیرد (یا همان متدهای تعریف شده در کنترلرها). همچنین از این URL‌ها پارامترهای این متدها یا اکشن‌ها نیز قابل استخراج است.

در متد `MapRoute`، اولین پارامتر تعریف شده، یک نام پیش فرض است و در ادامه اگر آدرسی را به فرم «یک چیزی اسلش یک چیزی اسلش یک چیزی» یافت، اولین قسمت آن را به عنوان نام کنترلر تفسیر خواهد کرد، دومین قسمت آن، نام متد عمومی موجود در کنترلر فرض شده و سومین قسمت به عنوان پارامتر ارسالی به این متد پردازش می‌شود. برای مثال از آدرس زیر اینطور می‌توان دریافت که:

```
http://hostname/home/about
```

Home نام کنترلی است که فریم ورک به دنبال آن خواهد گشت تا این درخواست رسیده را پردازش کند و `about` نام متدی عمومی در این کلاس است که به صورت خودکار فراخوانی می‌گردد. در اینجا پارامتر `id` ایی هم وجود ندارد. در یک برنامه امکان تعریف چندین و چند مسیریابی وجود دارد. پارامتر سوم متد `routes.MapRoute`، یک سری پیش فرض را تعریف می‌کند و این مورد همانجایی است که از اطلاعات آن جهت تعریف کنترلر پیش فرض استفاده کردیم. برای مثال به چهار آدرس زیر دقت نمائید:

```
http://localhost/
http://localhost/home
http://localhost/home/about
http://localhost/process/list
```

در حالت `http://localhost/`، هر سه مقدار پیش فرض مورد استفاده قرار خواهند گرفت چون سه جزئی `{controller}/{action}/{id}` که موتور مسیریابی به دنبال آن‌ها می‌گردد، در این آدرس وجود خارجی ندارد. بنابراین نام کنترلر پیش فرض در این حالت همان `Home` مشخص شده در پارامتر سوم متد `routes.MapRoute` خواهد بود و متد پیش فرض نیز `Index` و پارامتری هم به آن ارسال نخواهد شد. در حالت `http://localhost/home` نام کنترلر مشخص است اما دو جزء دیگر ذکر نشده‌اند، بنابراین مقادیر آن‌ها از پیش فرض‌های صریح ذکر شده در متد `routes.MapRoute` گرفته می‌شود. یعنی نام متد اکشن مورد پردازش، `Index` خواهد بود به همراه هیچ آرگومان خاصی. به علاوه باید خاطرنشان کرد که این مقادیر `case sensitive` یا حساس به بزرگی و کوچکی حروف نیستند. بنابراین مهم نیست که `http://localhost/hoMe` یا `http://localhost/HoMe` باشد یا `http://localhost/HOME` یا هر ترکیب دیگری. یا اگر آدرس رسیده `http://localhost/process/list` باشد، این مسیریابی پیش فرض تعریف شده قادر به پردازش آن می‌باشد. به این معنا که کنترلی به نام `Process` و هله سازی و سپس متدی به نام `List` در آن فراخوانی خواهند شد.

### یک نکته:

ترتیب `routes.MapRoute`های تعریف شده در اینجا مهم است و اگر اولین URL رسیده با الگوی تعریف شده مطابقت داشت، کار را تمام خواهد کرد و به سایر تعاریف نخواهد رسید. مثلاً اگر در اینجا یک مسیریابی دیگر را به نام `Process` تعریف کنیم:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "Process", // Route name
        "Process/{action}/{id}", // URL with parameters
        new { controller = "Process", action = "List", id = UrlParameter.Optional } //
Parameter defaults
    );

    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        new { controller = "Home", action = "Index", id = UrlParameter.Optional } // Parameter
defaults
    );
}
```

```
}
```

آدرسی به فرم `http://localhost/Process`، به صورت خودکار به کنترلر `Process` و متد عمومی `List` آن نگاشت خواهد شد و کار به مسیریابی بعدی نخواهد رسید.

## نظرات خوانندگان

نویسنده: Shima Besharat  
تاریخ: ۱۵:۴۲:۱۷ ۱۳۹۱/۰۱/۰۸

با سلام و تشکر از مهندس عزیز

یه خواهش:

امکان داره theme و Setting مربوط به Visual Studio خود را برای دانلود قرار بدهید. از هر سایتی setting گرفتم مثل setting شما در عکس بالا نشد. پیشاپیش ممنون.

نویسنده: وحید نصیری  
تاریخ: ۱۶:۵۶:۴۲ ۱۳۹۱/۰۱/۰۸

از اینجا می‌تونید دریافت کنید: ([^](#))

اگر از ویندوزهای بعد از XP استفاده می‌کنید که هیچ. اگر نه نیاز به فونت Consolas هم دارید که از سایت مایکروسافت قابل دریافت و نصب است: ([^](#))

نویسنده: Amin Noorivandi  
تاریخ: ۲۱:۵۶:۲۳ ۱۳۹۱/۰۱/۰۸

وحید جون داداش سرعتت خیلی کمه. ما الان تو ویو هستیم. تورو خدا یرسون جزوه بعدیتو

نویسنده: Shima Besharat  
تاریخ: ۰۰:۱۰:۲۲ ۱۳۹۱/۰۱/۰۹

بسیار ممنون از شما  
واقعاً بنیامین آهنگ دنیا دیگه مثل تو نداره رو برای شما خونده  
ممنون

نویسنده: Hossein Raziee  
تاریخ: ۰۰:۵۳:۰۵ ۱۳۹۱/۰۱/۰۹

اتفاقاً جناب نصیری خیلی سریع هم مینویسند.  
واقعا دستشون درد نکه.

نویسنده: Arezoo A  
تاریخ: ۲۱:۵۰:۳۹ ۱۳۹۱/۰۱/۱۲

یک دنیا تشکر. مثل همیشه عالیه

نویسنده: TazeKar  
تاریخ: ۱۲:۰ ۱۳۹۱/۰۳/۳۱

با سلام و تشکر فروان  
من هم مانند دوستان خیلی علاقمندم که از setting استفاده کنم که متاسفانه لینک دانلود خراب است لطفا بررسی کنید.  
ممنون ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۲:۴۰

من قبلا هم این پیام «خراب بودن» لینک دانلود رو جای دیگر هم دیده بودم. اینها خراب نیستند. روش CodePlex این است که بدون تنظیم Mime type خاصی، یک فایل را به درون مرورگر کاربر flush می‌کند. به این ترتیب کاربر فکر می‌کند که با یک سری اطلاعات باینری خراب شده طرف است که اینطور نیست. روی لینک کلیک راست کنید و گزینه save as رو انتخاب کنید. به این صورت درست دریافت خواهد شد.

نویسنده: maryam  
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۱:۱۹

سلام. با تشکر از زحمات شما. نحوه فعال کردن setting رو برای vs توضیح میدین. ممنون

نویسنده: TazeKar  
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۱:۲۷

واقعا ممنونم به چیز تازه یاد گرفتم هرچند اینجا پر از چیزهای برای یادگیری ولی این موارد خارج از بحث جالب ترش میکنه . قالب بندی جدید سایت هم خیلی خوبه کنترل و دسترسی در سایت رو برامون راحت تر کرده. بازم متشکرم.

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۲:۱۳

به این منو مراجعه کنید: Tools/Import and Export Settings

نویسنده: Ara  
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۶:۳۷

فلسفه کلاسهای abstract خیلی جالب بود  
ممنون که اینقدر ریز و دقیق مسئله رو باز کردین

نویسنده: vahid21  
تاریخ: ۱۳۹۱/۰۴/۱۹ ۱۵:۱۷

درود به همگی

از زحمات آقای مهندس نصیری واقعا ممنونم که بی دریغ دارن علم خودشون رو در اختیار هموطنانشون قرار میدن ، کمتر به همچین شخصیت هایی پیدا میشه!

من با اینکه اصلا با asp کار نکردم اما تصمیم گرفتم asp.net mvc رو با استفاده از این آموزش کامل ، فراگیرم و امیدوارم چون با asp کار نکردم به مشکل بر نخورم

نویسنده: مهدی  
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۳:۵۱

هنگامی که روی Action راست کلیک میکنم و درخواست ساختن View متناظر را می‌دهم، یک Dialoge باز می‌شود که می‌توان نام View را تعیین کرد. اگر نام View را متفاوت با نام Action قرار بدهم و View را بسازم و سپس پروژه را اجرا کنم، با درخواست آن Action، فریمورک به دنبال View ای با نام خود Action می‌گردد.  
سوال بنده این است که اگر لزوما نام Action و View باید یکسان باشد پس چرا MVC امکان تغییر نام View را به ما می‌دهد؟ آیا برای لینک کردن Action و View با نام متفاوت باید تنظیمات خاصی انجام داد؟



نویسنده: وحید نصیری  
تاریخ: ۱۳:۵۷ ۱۳۹۱/۰۵/۰۴

return View چندین و چند overload دارد. حالت پیش فرض آن استفاده از view ایی هم نام اکشن متد جاری است (اگر پارامتری ذکر نشود). اما امکان ذکر صریح viewName نیز وجود دارد. در اینجا حتی می شود مسیر دهی کامل هم انجام داد:

```
return View("~/Views/Wherever/SomeDir/MyView.cshtml")
```

نویسنده: امیر هاشم زاده  
تاریخ: ۲۰:۳۴ ۱۳۹۱/۰۵/۱۳

به مرور زمان و با بزرگ شدن پروژه نگهداری ارتباط اجزای MVC (کنترلر و View) و همچنین اکشن متدهای مربوط کنترلرها در Global.asax، توسعه و نگهداری رو سخت نمی‌کند؟

نویسنده: وحید نصیری  
تاریخ: ۲۰:۴۱ ۱۳۹۱/۰۵/۱۳

قرار نیست برای هر کنترلر و View ایی که تعریف می‌شود، در فایل global.asax.cs تغییری انجام شود. route پیش فرض آن در اکثر موارد جوابگو است.

مگر اینکه بخواهید redirect های خاصی را تعریف کنید. مثلا لینک های تگ ها در بلاگر به این شکل بودند و هستند :

<http://site/search/label/tagname>

برای اینکه این نوع لینک های رسیده و bookmark شده یا ثبت شده در سایت های مختلف به آدرس جدید <http://site/tag/name> هدایت شوند، می شود یک route برای آن نوشت:

```
routes.MapRoute(
    "old_blogger", // Route name
    "search/label/{name}", // URL with parameters
    new { controller = "Tag", action = "Index", name = UrlParameter.Optional } // Parameter
    defaults
);
```

به این ترتیب به صورت خودکار آدرس های قدیمی به آدرس های جدید (کنترلر Tag سایت) نگاشت خواهند شد و از این دست موارد. خلاصه فایل global.asax.cs نیاز به آنچنان تغییری ندارد و پیش فرض های آن در بسیاری از موارد کافی هستند.

نویسنده: امیر هاشم زاده  
تاریخ: ۲۰:۴۹ ۱۳۹۱/۰۵/۱۳

پس در واقع فقط با داشتن یک route پیش فرض، سایر اکشن متدها با url درخواستی لینک می شوند؟

نویسنده: وحید نصیری  
تاریخ: ۲۰:۵۳ ۱۳۹۱/۰۵/۱۳

در متن توضیح دادم. route پیش فرض به صورت خیلی کلی زیر تعریف شده:

```
{controller}/{action}/{id}
```

یعنی تطابق با «یک چیزی اسلش یک چیزی اسلش یک چیزی» و اکثر کاربردها را پوشش می دهد. یک چیزی اول می شود نام کنترلر. یک چیزی دوم می شود نام اکشن متدی در آن کنترلر (در صورت وجود) و الی آخر. بنابراین نیازی نیست به ازای هر کنترلر جدید یک route نوشت؛ یا به ازای هر متدی در آن به همین ترتیب.

نویسنده: مهمن  
تاریخ: ۱۳:۳۹ ۱۳۹۱/۰۵/۱۹

خیلی ممنون

نویسنده: امیر  
تاریخ: ۲۳:۳۸ ۱۳۹۱/۰۷/۲۸

سلام

من یک سایت دیدم که با mvc نوشته شده یکی از لینک‌های سایت به این شکل بودش

لینک‌های مرتبط <http://site.ir/>

میخواستم بدونم این لینک‌های مرتبط چی هستش؟ قاعدتا اسم کنترلر نیستش چگونه اینکار صورت گرفته؟  
مرسی

نویسنده: وحید نصیری  
تاریخ: ۲۳:۵۰ ۱۳۹۱/۰۷/۲۸

- در انتهای قسمت 9 در مورد لینک‌های فارسی توضیح دادم: ( [^](#) )
- همچنین شما در MVC می‌تونید برای یک آدرس خاص route بنویسید که در پشت صحنه به کنترلر خاصی [هدایت شود](#)
- مطلب « [سازگار کردن لینک‌های قدیمی یک سایت با ساختار جدید آن در ASP.NET MVC](#) » توضیحات بیشتری به همراه دارد

نویسنده: اوبالیت  
تاریخ: ۱۳:۲ ۱۳۹۱/۱۱/۰۴

درود؛ دو سوال داشتم.

- 1) هنگامی که من می‌خواستم یک View رو ایجاد کنم، فایل ایجاد شده برای من Index.ascx بود نه Index.cshtml. علت چی می‌تونه باشه؟
- 2) چرا وقتی در صفحه Index شروع به کد نویسی کردم، VS برای من ادامه دستورات رو نشون نمی‌داد؟ یعنی وقتی Ctrl+Space می‌زدم. من دستورات رو خودم نوشتم، یعنی:

```
@{
    ViewBag.....
```

مشکلی وجود دارد یا جز ویژگی‌های MVC محسوب میشود؟

نویسنده: وحید نصیری  
تاریخ: ۱۳:۳۸ ۱۳۹۱/۱۱/۰۴

- بستگی داره نوع View Engine را هنگام آغاز پروژه روی web forms قرار دادید یا Razor انتخاب شده. اگر فایل ascx تولید میشه یعنی web forms انتخاب شده.
- اگر نوع View Engine انتخابی web forms باشد در حالت شما دستورات @ مربوط به Razor معنایی نخواهند داشت. ضمناً ViewBag یک شیء Dynamic است و کلاً برای آن intellisense وجود ندارد.

نویسنده: عدالت  
تاریخ: ۱۹:۴۶ ۱۳۹۲/۰۹/۱۲

با سلام؛ چند تا سوال دارم اونم اینک:

- 1- اگر در قرار داد ما در rout که این باشه {controller}/{action}/{id} یکی از اون‌ها رو ننویسیم چه اتفاقی میافته؟

2- اگر قرارداد درست بود و ما در قسمت دوم مسیر پیش فرض یکی از بخش‌ها رو مشخص نکرده باشیم چه اتفاقی می‌افته؟  
مثلاً اگر در بخش دوم که `action="index"` بود اگر ذکر نشود بین تعداد زیادی از متدهای موجود در کنترلر کدام یکی انتخاب خواهد شد؟

```
routes.MapRoute(
    name: "about",
    url: "{controller}/{action}/{id}", defaults: new { Controller = "Home", id =
    UrlParameter.Optional }

);
```

اولویتی برای اینکار وجود دارد؟  
آیا با این کار مسیر دهی اشتباه صورت گرفته؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۱:۵۰

- یک مسیریابی جدید تعریف می‌شود.
- خطای 404 دریافت می‌کنید. متد دیگری انتخاب نمی‌شود. اولویتی برای انتخاب اکشن متد بعدی وجود ندارد؛ بجز مواردی که صریحاً به صورت مقدار پیش فرض مسیریابی مشخص شدن.
- اولویت فقط در مورد خود مسیریابی‌ها وجود دارد. برای توضیحات بیشتر مراجعه کنید به مطلب:  
« [چطور مسیریابی‌های ASP.NET MVC را دیباگ کنیم؟](#) »

نویسنده: ع طاهری  
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۶:۲۷

سلام؛ تو کنترلر متدی دارم که ورودی آن `int` و خروجی `string` ای را که از دیتابیس می‌گیرد را بر میگرداند. مشکلم اینه که ورودی متد رو باید از `view` و بر اساس مقداری که کاربر در حین اجرای برنامه تو `textbox` وارد میکنه بگیرم و خروجی متد را در `view` به کاربر نشون بدم. چیکار باید بکنم؟ البته مقدار عددی یکی از `property`های `viewmodel` هست. اما رشته نه!

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۸:۴۹

بررسی نحوه انتقال اطلاعات از یک کنترلر به `View`های مرتبط با آن [در قسمت پنجم](#) مطرح شده؛ آشنایی با روش‌های مختلف ارسال اطلاعات یک درخواست به کنترلر [در قسمت دهم](#) (چون یک سری پیشنهاد لازم داشته). روش دیگر انجام این کارها استفاده از Ajax است که [در قسمت 21](#) به آن پرداخته شده.