

فرض کنید یک پوشه Export در ریشه سایت دارید که حاوی تعدادی فایل PDF عمومی است.  
**سؤال:** آیا می‌شود دسترسی به فایل‌های قرار گرفته در این پوشه عمومی را کنترل کرد؟ به نحوی که فقط کاربران عضو سایت پس از اعتبارسنجی بتوانند آن‌ها را دریافت کنند؟  
 پاسخ: شاید عنوان کنید که می‌توان از تگ location در فایل web.config برای اینکار استفاده کرد:

```
<location path="Export">
  <system.web>
    <authorization>
      <deny users="?>" />
    </authorization>
  </system.web>
</location>
```

این تنظیمات هیچ اثری بر روی فایل‌های استاتیک PDF ندارند؛ چون در IIS 6 از موتور ASP.NET رد نخواهند شد. مگر اینکه این نوع پسوندها به aspnet\_isapi.dll انتساب داده شوند. در IIS 7 به بعد این وضع بهبود یافته است. اگر تنظیم runAllManagedModulesForAllRequests در وب کانفیگ برنامه به true تنظیم شده باشد و برنامه در حالت Integrated pipeline بجای Classic mode اجرا شود، امکان مدیریت فایل‌های استاتیک نیز در برنامه‌های ASP.NET [وجود دارد](#) .

**سؤال:** آیا راه حلی وجود دارد که بتوان فایل‌های استاتیک را صرفنظر از نوع، نگارش و حالت اجرای IIS توسط موتور ASP.NET مدیریت کرد؟  
 پاسخ: بلی. در ASP.NET MVC با تنظیم یک سطر ذیل، اینکار انجام می‌شود:

```
public static void RegisterRoutes(RouteCollection routes)
{
    // ...
    routes.RouteExistingFiles = true;
    // ...
}
```

توضیحات:

RouteCollection در ASP.NET MVC به کمک امکانات MapPathBasedVirtualPathProvider خود، ابتدا درخواست رسیده را بررسی می‌کند. اگر این درخواست به یک فایل عمومی اشاره کند، کل سیستم مسیریابی را غیرفعال می‌کند. اما با تنظیم RouteExistingFiles دیگر این بررسی صورت نخواهد گرفت (به عبارتی در بالا بردن سرعت نمایشی سایت نیز تاثیر گذار خواهد بود؛ چون یکی از بررسی‌ها را حذف می‌کند).

**ایجاد کنترلری به نام پوشه‌ای که قرار است محافظت شود**

نام پوشه قرار گرفته در ریشه سایت، Export است. بنابراین برای هدایت درخواست‌های رسیده به آن (پس از تنظیم فوق)، نیاز است یک کنترلر جدید را به نام Export نیز ایجاد کنیم:

```
using System.IO;
using System.Web.Mvc;

namespace Mvc4RouteExistingFiles.Controllers
{
    public class ExportController : Controller
    {
        public ActionResult Index(string id)
        {
            if (string.IsNullOrEmpty(id))
            {
                return Redirect("/");
            }
        }
    }
}
```

```

    }
    var fileName= Path.GetFileName(id);
    var path = Server.MapPath("~/export/" + fileName);
    return File(path, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
}
}
}

```

البته بدیهی است در اینجا می‌توان فیلتر `Authorize` را به کل کنترلر اعمال کرد، یا هر تنظیم دیگری که نیاز است. برای اینکه بررسی کنیم، آیا واقعا فایل‌های استاتیک قرار گرفته در پوشه `Export` به این کنترلر هدایت می‌شود یا خیر، یک breakpoint را بر روی سطر اول اکشن متد `Index` قرار می‌دهیم. برنامه را اجرا کنید ... کار نخواهد کرد، زیرا مسیر یک فایل فرضی به صورت ذیل:

`http://localhost/export/test.pdf`

به اکشن متد `Index` کنترلر `Export`، نگاشت نخواهد شد (index در این مسیر ذکر نشده است). برای حل این مشکل فقط کافی است مسیر یابی متناظری را تعریف کنیم:

```

using System.Web.Mvc;
using System.Web.Routing;

namespace Mvc4RouteExistingFiles
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.RouteExistingFiles = true;

            routes.MapRoute(
                name: "ExportRoute",
                url: "Export/{id}",
                defaults: new { controller = "Export", action = "Index", id = UrlParameter.Optional }
            );

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
            );
        }
    }
}

```

در اینجا `ExportRoute` را مشاهده می‌کنید که به آدرس‌هایی به فرم `Export/id` پاسخ می‌دهد. در این حالت به صورت خودکار با توجه به تنظیمات انجام شده، اکشن متدی که انتخاب می‌شود همان `Index` خواهد بود و نیازی به ذکر صریح آن نخواهد بود. اینبار اگر برنامه را اجرا کنیم، breakpoint ما کار خواهد کرد:

```
1 using System.IO;
2 using System.Web.Mvc;
3
4 namespace Mvc4RouteExistingFiles.Controllers
5 {
6     public class ExportController : Controller
7     {
8         public ActionResult Index(string id)
9         {
10             if (string.IsNullOrEmpty(id))
11             {
12                 return Redirect("/");
13             }
14
15             var fileName = Path.GetFileName(id);
16             var path = Server.MapPath("~/export/" + fileName);
17             return File(path, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
18         }
19     }
20 }
```

### تنظیمات ثانویه پس از فعال سازی RouteExistingFiles

در این حالت با فعال سازی مسیریابی فایل‌های موجود، دیگر هیچ فایل استاتیکی به صورت معمول در اختیار کاربران قرار نخواهد گرفت و اگر همانند توضیحات قبل برای آن‌ها کنترلر جداگانه‌ای را تهیه نکنیم، عملاً سایت از کار خواهد افتاد. برای رفع این مشکل، در ابتدای متد RegisterRoutes فوق، تنظیمات ذیل را اضافه کنید تا پوشه‌های content، scripts و همچنین یک سری فایل با پسوند مشخص، همانند سابق و مستقیماً توسط سرور ارائه شوند؛ در غیراینصورت کاربر پیغام 404 را پس از درخواست آن‌ها، دریافت خواهد کرد:

```
routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
routes.IgnoreRoute("content/{*pathInfo}");
routes.IgnoreRoute("scripts/{*pathInfo}");
routes.IgnoreRoute("favicon.ico");
routes.IgnoreRoute("{resource}.ico");
routes.IgnoreRoute("{resource}.png");
routes.IgnoreRoute("{resource}.jpg");
routes.IgnoreRoute("{resource}.gif");
routes.IgnoreRoute("{resource}.txt");
```

## نظرات خوانندگان

نویسنده: خوشقدم  
تاریخ: ۱۶:۰ ۱۳۹۲/۰۴/۱۲

سلام و خسته نباشید

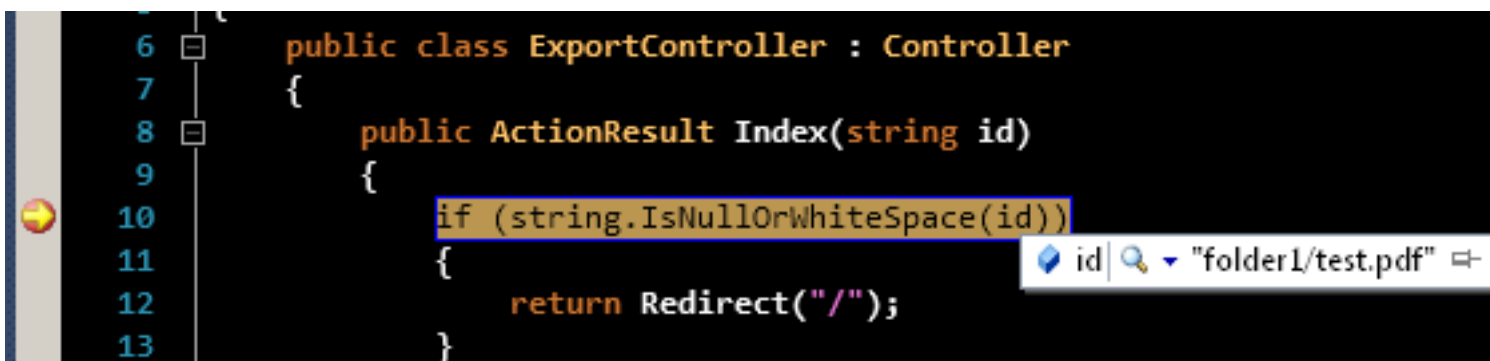
اکشن تعریف شده تنها برای مقدار Id اجرا خواهد شد اما در صورتی که فولدر Export شامل زیرشاخه باشد چه کار باید کرد

نویسنده: وحید نصیری  
تاریخ: ۱۸:۳۰ ۱۳۹۲/۰۴/۱۲

یک ستاره قبل از id قرار بدید:

```
routes.MapRoute(
    name: "ExportRoute",
    url: "Export/{*id}",
    defaults: new { controller = "Export", action = "Index", id = UrlParameter.Optional }
);
```

نتیجه آن:



نویسنده: حمید چگینی  
تاریخ: ۱۲:۳۷ ۱۳۹۲/۰۹/۱۵

بدون اینکه تنظیمات قسمت مسیریابی رو انجام بدم این مسیر باز میشه:

<http://localhost:1431/Export/api.pdf>

و در حالت پیش فرض هم فایل‌های موجود در فولدرهای Scripts و Content با تایپ مسیر آنها تو Url همگی قابل دسترس هستند.

- تو درک این عبارت مشکل دارم. ممکنه کمی در این باره توضیح بدین:

```
routes.IgnoreRoute( "{resource}.axd/{*pathInfo}" );
```

ممنون

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۲:۴۳

- بله. این‌ها فایل استاتیک هستند و همانطور که در بحث عنوان شده، به صورت مستقیم و مستقل، توسط IIS مدیریت می‌شوند. موضوع بحث جاری، هدایت این فایل‌های استاتیک، به یک برنامه ASP.NET MVC است؛ پیش از اینکه IIS راسا تصمیم‌گیری کند. - یعنی از هر مسیری که پسوند axd داشت با هر جزء دیگری از مسیر پس از آن، صرفنظر شود.

نویسنده: علی  
تاریخ: ۱۳۹۲/۰۹/۲۹ ۱۲:۴۳

سلام من هر کاری می‌کنم وقتی تو مرورگر درخواست فایل استاتیکی رو میدم بلافاصله خطای 404 میده و کنترلر اصلا اجرا نمیشه. در متد registerRoutes همه آدرس رو تعریف کرده ام و routeexistingfiles رو هم true کردم یه کنترلر و یه متد هم طبق برنامه بالا تعریف کرده ام ولی به محض درخواست فایل خطای 404 میده در اصل اون فایل وجود نداره ولی من میخوام قبل از اینکه وجود فایل رو بررسی کنه ابتدا کنترلر من رو اجرا کنه همچین چیزی ممکنه؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۹/۲۹ ۱۳:۵۵

بله. به شرطی که برای آن [route مخصوص](#) بنویسید تا ASP.NET MVC بداند درخواست رسیده را باید به کجا ارسال کند. ضمن اینکه اگر route جدید شما اجرا نمی‌شود، باید مسیریابی‌های موجود را جهت رفع تداخل [دیب‌گ کنید](#).

نویسنده: ایاک  
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۲:۳

با سلام و تشکر.  
متد Application\_AuthenticateRequest من به ازای هر تقاضا برای فایل‌های استاتیک اجرا می‌شود. آیا باید اجرا شود و اگر نه، به چه نحوی باید آنرا کنترل کرد تا وارد این مرحله نشود؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۲:۱۸

سؤال و هدف اصلی بحث جاری این است: «آیا می‌شود دسترسی به فایل‌های قرار گرفته در این پوشه عمومی را کنترل کرد؟ به نحوی که فقط کاربران عضو سایت پس از اعتبارسنجی بتوانند آن‌ها را دریافت کنند؟»  
یعنی تمام اینکارها انجام شد تا بتوان دریافت فایل‌های استاتیک را تحت کنترل کامل برنامه و اعتبارسنجی آن قرار داد. اگر نیازی نیست، خوب، مباحث آنرا پیاده سازی نکنید. همچنین مانند IgnoreRoute نوشته شده در انتهای بحث برای پوشه اسکریپت‌ها یا CSSها، پوشه‌ی مدنظر را از سیستم مسیریابی خارج کنید.

نویسنده: علی محبی  
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۷:۲۸

با وجود اضافه کردن `routes.RouteExistingFiles = true` و تنظیم مسیره‌ی RouteConfig اما با صدا زدن <http://localhost/export/test> جواب می‌گیرم ولی با <http://localhost/expor/test.jpg> خطای 404 دریافت می‌کنم

دستور IgnoreRoute مربوطه را هم اعمال کردم.

من از mvc5 استفاده می‌کنم، مطلب فوق در این ورژن هم صادق است؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۲/۲۵ ۲۱:۲۱

فرقی نمی‌کند: [Mvc5RouteExistingFiles.zip](#)

فقط اگر قرار است تمام فایل‌های استاتیک پوشه‌ی export، منهای تصاویر jpg آن route شوند، باید یک سطر زیر را هم اضافه کنید:

```
routes.IgnoreRoute("export/{resource}.jpg");
```

نویسنده: ابراهیم

تاریخ: ۲۳:۲۴ ۱۳۹۳/۰۱/۲۱

فایلهایی که به این صورت دانلود می‌شن قابلیت resume ندارند ، راهی برای resume وجود داره ؟

نویسنده: وحید نصیری

تاریخ: ۲۳:۲۹ ۱۳۹۳/۰۱/۲۱

» [File ActionResult](#) هایی با قابلیت از سرگیری مجدد برای ASP.NET MVC «