

درگیر شدن با سایت‌های دیگر که چرا مطالب ما را کپی کرده‌اید نهایتاً بجز فرسایش عصبی حاصل دیگری را به همراه ندارد. اساساً زمانیکه مطلبی را به صورت باز در اینترنت انتشار می‌دهید، قید کپی شدن یا نشدن آن را باید زد. اما ... می‌توان همین سایت‌ها را تبدیل به تبلیغ‌کننده‌های رایگان کار خود نمود که در ادامه نحوه انجام آن را در یک برنامه ASP.NET MVC بررسی خواهیم کرد:

الف) نیاز است ارائه تصاویر تحت کنترل برنامه باشند.

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

namespace MvcWatermark.Controllers
{
    public class HomeController : Controller
    {
        const int ADay = 86400;

        public ActionResult Index()
        {
            return View();
        }

        [OutputCache(VaryByParam = "fileName", Duration = ADay)]
        public ActionResult Image(string fileName)
        {
            fileName = Path.GetFileName(fileName); // تمیز سازی امنیتی است
            var rootPath = Server.MapPath("~/App_Data/Images");
            var path = Path.Combine(rootPath, fileName);
            if (!System.IO.File.Exists(path))
            {
                var notFoundImage = "notFound.png";
                path = Path.Combine(rootPath, notFoundImage);
                return File(path, MediaTypeNames.Image.Gif, notFoundImage);
            }
            return File(path, MediaTypeNames.Image.Gif, fileName);
        }
    }
}
```

در اینجا یک کنترلر را مشاهده می‌کنید که در اکشن متد Image آن، نام یک فایل دریافت شده و سپس این نام در پوشه App_Data/Images جستجو گردیده و نهایتاً در مرورگر کاربر Flush می‌شود. از آنجائیکه الزامی ندارد fileName، واقعاً یک fileName صحیح باشد، نیاز است توسط متد استاندارد Path.GetFileName این نام دریافتی اندکی تمیز شده و سپس مورد استفاده قرار گیرد. همچنین جهت کاهش بار سرور، از یک OutputCache به مدت یک روز نیز استفاده گردیده است. نحوه استفاده از این اکشن متد نیز به نحو زیر است:

```

```

ب) آیا فراخوان تصویر ما را مستقیماً در سایت خودش قرار داده است؟

```
private bool isEmbeddedIntoAnotherDomain
{
    get
```

```
{
    return this.HttpContext.Request.UrlReferrer != null &&
!this.HttpContext.Request.Url.Host.Equals(this.HttpContext.Request.UrlReferrer.Host,
StringComparison.InvariantCultureIgnoreCase);
}
```

در ادامه توسط خاصیت سفارشی `isEmbeddedIntoAnotherDomain` درخواهیم یافت که درخواست رسیده، از دومین جاری صادر شده است یا خیر. اینکار توسط بررسی `UrlReferrer` ارسال شده توسط مرورگر صورت می‌گیرد. اگر `Host` این `UrlReferrer` با `Host` درخواست جاری یکی بود، یعنی تصویر از سایت خودمان فراخوانی شده‌است.

ج) افزودن خودکار Watermark در صورت کپی شدن در سایتی دیگر

```
private byte[] addWaterMark(string filePath, string text)
{
    var image = new WebImage(filePath);
    image.AddTextWatermark(text);
    return image.GetBytes();
}
```

کلاسی در فضای نام `System.Web.Helpers` وجود دارد به نام `WebImage` که کار افزودن Watermark را بسیار ساده کرده است. نمونه‌ای از نحوه استفاده از آنرا در متد فوق ملاحظه می‌کنید. اما ... پس از امتحان تصاویر مختلف ممکن است گاهی با خطای زیر مواجه شویم:

A Graphics object cannot be created from an image that has an indexed pixel format.

مشکل از اینجا است که تصاویر با فرمت ذیل برای انجام کار Watermark پشتیبانی نمی‌شوند:

```
PixelFormatUndefined
PixelFormatDontCare
PixelFormat1bppIndexed
PixelFormat4bppIndexed
PixelFormat8bppIndexed
PixelFormat16bppGrayScale
PixelFormat16bppARGB1555
```

اما می‌توان تصویر دریافتی را ابتدا تبدیل به BMP کرد و سپس Watermark دار نمود:

```
private byte[] addWaterMark(string filePath, string text)
{
    using (var img = System.Drawing.Image.FromFile(filePath))
    {
        using (var memStream = new MemoryStream())
        {
            using (var bitmap = new Bitmap(img))//avoid gdi+ errors
            {
                bitmap.Save(memStream, ImageFormat.Png);
                var webImage = new WebImage(memStream);
                webImage.AddTextWatermark(text, verticalAlign: "Top", horizontalAlign: "Left",
fontColor: "Brown");
                return webImage.GetBytes();
            }
        }
    }
}
```

در اینجا نمونه اصلاح شده متد `addWaterMark` فوق را بر اساس کار با تصاویر `bmp` و سپس تبدیل آن‌ها به `png`، ملاحظه می‌کنید. به این ترتیب دیگر به خطای یاد شده بر نخواهیم خورد.

در ادامه، قسمت آخر کار، اعمال این مراحل به اکشن متد Image است:

```
if (isEmbeddedIntoAnotherDomain)
{
    var text = Url.Action(actionName: "Index", controllerName: "Home", routeValues: null,
protocol: "http");
    var content = addWaterMark(path, text);
    return File(content, MediaTypeNames.Image.Gif, fileName);
}
return File(path, MediaTypeNames.Image.Gif, fileName);
```

در اینجا اگر تشخیص داده شود که تصویر، در دومین دیگری لینک شده است، آدرس سایت ما به صورت خودکار در بالای تصویر درج خواهد شد.

کدهای نهایی این کنترلر را از اینجا می‌توانید دریافت کنید:

[HomeController.cs](#)

به همراه نمونه تصویری که استثنای یاد شده را تولید می‌کند؛ جهت آزمایش بیشتر:

[EFStra08.gif](#)

نظرات خوانندگان

نویسنده: محسن جمشیدی
تاریخ: ۸:۵۷ ۱۳۹۲/۰۲/۱۹

اتفاقا دیروز دیدم در سایتی مطالب این سایت استفاده شده بود که برایم خوشایند نبود. توجهم به تصاویر جلب شد که نام dotnettips درج شده بود فکر کردم که جدیداً این امکان اضافه شده و بعد از بررسی متوجه شدم که این اتفاق در سایتهای دیگر می افتد برایم جالب بود

نویسنده: وحید نصیری
تاریخ: ۸:۲۷ ۱۳۹۲/۰۳/۱۴

جهت تکمیل بحث، [با استفاده از IIS Url Rewrite](#) نیز می توان شبیه چنین کاری را انجام داد.

نویسنده: حمید حسین وند
تاریخ: ۰:۲۳ ۱۳۹۳/۰۲/۱۰

سلام
آیا میشه همچین کاری رو توی asp.net web form کرد؟
یا اصلاً به کاری میشه کرد وقتی تصویر رو از طریق همین redactor (فایل ashx) آپلود میکنیم همون لحظه لوگویی یا متنی رو بهش اضافه کنه؟

نویسنده: وحید نصیری
تاریخ: ۰:۳۲ ۱۳۹۳/۰۲/۱۰

کدهای فوق قابل تبدیل به یک Generic handler وب فرمها (فایل های ashx) هستند. در اینجا بجای Url.Action فوق برای مقدار دهی img src، خواهید داشت `pic.ashx?filename=tst.png`.
ضمناً ASP.NET MVC سورس باز است و کدهای متد AddTextWatermark آن را [در اینجا](#) می توانید مطالعه کنید.