



گوگل خلاصه نتایج Indexing یک سایت را توسط ابزاری به نام [Google webmaster tools](https://www.google.com/webmaster/tools/) در اختیار علاقمندان قرار می‌دهد. Bing نیز [چنین ابزاری](#) را تدارک دیده است. به آمارهای خطای حاصل از سایت جاری که دقت می‌کردم یک نکته آن جالب بود: «محتوای تکراری»

▼ [ASP.Net در Google Analytics استفاده از - .NET Tips](#)  
 [/Post/33/--Google-Analytics--ASP-Net](#)  
 [/post/33/google-analytics-در-asp-net-استفاده-از](#)

mydomain.com/Home/Index  
 mydomain.com/home/index  
 mydomain.com/Home/index  
 mydomain.com/home/Index

همانطور که ملاحظه می‌کنید، گوگل به کوچکی و بزرگی حروف بکار رفته در لینک‌ها حساس است. هرچند 4 لینک فوق به یک صفحه اشاره می‌کنند، اما گوگل 4 بار آن‌ها را ایندکس خواهد کرد و نهایتاً به صورت یک خطای «محتوای تکراری» در گزارشات SEO آن ظاهر خواهد شد (به همراه کاهش رتبه SEO سایت).

## راه حل

برای حل این مساله دو نکته باید در نظر گرفته شود:

الف) هدایت دائمی (Redirect permanent) صفحات قدیمی به صفحاتی جدید، با آدرس lowercase

```
using System.Globalization;
using System.Web;
using System.Web.Mvc;

namespace WebToolkit
{
    public class ForceWww : ActionFilterAttribute
    {
        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            modifyUrlAndRedirectPermanent(filterContext);
            base.OnActionExecuting(filterContext);
        }

        private static void modifyUrlAndRedirectPermanent(ActionExecutingContext filterContext)
        {
            if (canIgnoreRequest(filterContext))
                return;

            var absoluteUrl =
                HttpUtility.UrlDecode(filterContext.HttpContext.Request.Url.AbsoluteUri.ToString(Culture
                Info.InvariantCulture));
            var absoluteUrlToLower = absoluteUrl.ToLowerInvariant();

            absoluteUrlToLower = forceWwwAndLowercase(filterContext, absoluteUrlToLower);
            absoluteUrlToLower = avoidTrailingSlashes(filterContext, absoluteUrlToLower);

            if (!absoluteUrl.Equals(absoluteUrlToLower))
            {
                filterContext.Result = new RedirectResult(absoluteUrlToLower, permanent: true);
            }
        }
    }
}
```

```

    }
}

private static string avoidTrailingSlashes(ActionExecutingContext filterContext, string
absoluteUrlToLower)
{
    if (!isRootRequest(filterContext) && absoluteUrlToLower.EndsWith("/"))
        return absoluteUrlToLower.TrimEnd(new[] { '/' });

    return absoluteUrlToLower;
}

private static bool isRootRequest(ActionExecutingContext filterContext)
{
    return filterContext.RequestContext.HttpContext.Request.Url.AbsolutePath == "/";
}

private static bool canIgnoreRequest(ActionExecutingContext filterContext)
{
    return filterContext.IsChildAction ||
        filterContext.HttpContext.Request.IsAjaxRequest() ||
        filterContext.RequestContext.HttpContext.Request.Url.AbsoluteUri.Contains("?");
}

private static string forceWwwAndLowercase(ActionExecutingContext filterContext, string
absoluteUrlToLower)
{
    if (isLocalRequet(filterContext))
        return absoluteUrlToLower;

    if (absoluteUrlToLower.Contains("www"))
        return absoluteUrlToLower;

    return absoluteUrlToLower.Replace("http://", "http://www.")
        .Replace("https://", "https://www.");
}

private static bool isLocalRequet(ActionExecutingContext filterContext)
{
    return filterContext.RequestContext.HttpContext.Request.IsLocal;
}
}
}

```

کلاس فوق، نگارش تکمیل شده [ForceWww](#) که پیشتر در این سایت دیده‌اید. توسط آن سه بررسی مختلف بر روی لینک جاری در حال پردازش صورت خواهد گرفت:

- تمام آدرس‌های سایت باید `www` داشته باشند؛ تا آدرس‌های آن یکنواخت شده و خصوصا مشکلات لاگین و نوشته شدن کوکی‌ها به ازای آدرس‌های مختلف و سر درگمی کاربران کاهش یابد.
- اگر آدرس جاری `lowercase` نباشد، تبدیل به نمونه `lowercase` شده و درخواست کننده، به آدرس جدید هدایت می‌شود. این مورد خصوصا جهت موتورهای جستجو برای تصحیح نتایج آن‌ها بسیار مفید است.
- اسلش انتهایی لینک‌ها در صورت وجود حذف خواهد شد. این مورد نیز در کاهش تعداد خطاهای «محتوای تکراری» مؤثر است.
- اگر آدرسی، کوئری استرینگ داشته باشد از آن صرفنظر خواهد شد؛ زیرا ممکن است اطلاعات موجود در آن به کوچکی و بزرگی حروف حساس باشند.

### ب) کاهش بار سایت توسط تولید خودکار `Url`هایی که در بدو امر `lowercase` هستند

برای پیاده سازی این مطلب می‌توان از پروژه سورس باز « [LowercaseRoutesMVC](#) » استفاده کرد. سه فایل `cs` دارد که می‌توانید به پروژه خود اضافه کنید. پس از آن، هرجایی در پروژه خود `routes.MapRoute` دارید تبدیل کنید به `routes.MapRouteLowercase`. به این ترتیب به صورت خودکار تمام `Url`های تولید شده توسط `HTML helpers` توکار `ASP.NET MVC` (و نه `Url`هایی که دستی نوشته شده‌اند)، در حین درج در صفحه به صورت `lowercase` ظاهر خواهند شد (صرفنظر از اینکه نام‌های کنترلرها و یا اکشن متدهای تعریف شده `camel case` هستند یا خیر). مزیت این مساله کاهش یک مرحله `Redirect` است که در قسمت الف ذکر شد. در این کتابخانه کمکی نیز از آدرس‌هایی که دارای کوئری استرینگ باشند، صرفنظر می‌شود.

## نظرات خوانندگان

نویسنده: امیرحسین مرجانی  
تاریخ: ۱۳۹۱/۱۰/۱۵ ۲۳:۲۲

سلام  
ممنونم ، مطلب مفیدی بود.  
اگر ممکنه بیشتر روی بحث SEO مطلب منتشر کنید.

نویسنده: یاسر مرادی  
تاریخ: ۱۳۹۱/۱۰/۱۶ ۱۰:۳۲

بهتر نبود برای تشخیص این که Request دارای Query String است یا خیر، قسمت Query String آن را بررسی می کردید که مقدار دارد یا خیر ؟  
تا این که در کل کوئری دنبال علامت ؟ بگردید ؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۰/۱۶ ۱۱:۱

- تشخیص وجود [کوئری استرینگ](#) در یک Url، حالت دیگری نمی تونه داشته باشه.  
- ضمنا یک Url می تونه کوئری استرینگ با چندین name-value داشته باشه. ضرورتی به بررسی تمام موارد و کند کردن کار نیست که آیا مقدار دارند همگی یا خیر.

نویسنده: توحید عزیزی  
تاریخ: ۱۳۹۱/۱۰/۱۶ ۲۲:۳۷

این پاسخ در stackoverflow.com به نقل از اسکات گو هم بد نیست: <http://stackoverflow.com/a/170955/538387>

نویسنده: محسن  
تاریخ: ۱۳۹۱/۱۰/۱۷ ۰:۳۳

راه مناسبی نیست. چون قید ( [IRouteConstraint](#) ) اعمال می کنه و سبب می شه که خیلی از url ها در سایت پردازش نشن.

نویسنده: جلال  
تاریخ: ۱۳۹۱/۱۰/۱۸ ۲:۵۵

سلام  
در تابع forceWwwAndLowerCase بهتره if مربوط به www به این شکل تغییر کنه:

```
if (absoluteUrlToLower.Contains("http://www.") || absoluteUrlToLower.Contains("https://www."))
    return absoluteUrlToLower;
```

اینطوری اگه توی آدرس، Controller یا Action دارای www باشه، مشکلی پیش نیاد. البته نکته دیگه درباره آدرس های دارای query string این که چرا تا قبل از ؟ رو به حروف کوچک تبدیل نکنیم؟ دلیل خاصی داره؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۱۰/۱۸ ۸:۴۷

- این یک کار سورس باز هست. مطابق نیاز خودتون تغییرش بدید و استفاده کنید. برای استفاده ای که من دارم تنظیمات آن کافی

است.

- در کار من قسمت‌هایی که کوئری استرینگ دارند عمومی نیستند و نیاز به اعتبارسنجی دارند. به همین جهت تبدیل آن‌ها برای بحث SEO اهمیتی نخواهند داشت. خصوصا قسمت کوئری استرینگ آن‌ها نباید تغییری کند چون در رمزگشایی اطلاعات از آن استفاده می‌شود.

نویسنده: وحید نصیری  
تاریخ: ۱۲:۵۵ ۱۳۹۱/۱۰/۲۶

تکمیل بحث:

گوگل به تنظیم [canonical url](#) هم نیاز دارد. بنابراین تنظیم ذیل به کلاس فوق اضافه شد:

```
filterContext.Controller.ViewBag.CanonicalUrl = absoluteUrlToLower;
```

و بعد در فایل layout برنامه خواهیم داشت:

```
<link rel="canonical" href="@ViewBag.CanonicalUrl"/>
```

نویسنده: وحید نصیری  
تاریخ: ۰:۲ ۱۳۹۱/۱۲/۰۳

یک نکته تکمیلی دیگر:

```
<a href="/contactus?name=وحید نصیری" rel="nofollow">ارسال پیام خصوصی</a>
```

اگر در سایت لینک‌هایی دارید که نیاز به اعتبارسنجی و لاگین دارند، این‌ها را با [rel=nofollow](#) مشخص کنید تا توسط گوگل ایندکس نشوند و تمام آن‌ها به یک صفحه تکراری (از دیدگاه گوگل) ختم نگردند.

نویسنده: وحید نصیری  
تاریخ: ۱۴:۷ ۱۳۹۲/۰۳/۰۱

نکته: پشتیبانی توکار از ASP.NET MVC از Lowercase generated URLs :

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

نویسنده: vahid  
تاریخ: ۱۳:۲۷ ۱۳۹۲/۱۱/۰۶

سلام

من با دات نت 4.5 نتونستم از این امکان استفاده کنم. در لینک زیر می‌گه این باگ مال دات نت 4.5 و در 5 درست شده

[stackoverflow](#)

آیا راهی هست بشه این باگ رفع کرد (بدون ابزار)

نویسنده: وحید نصیری  
تاریخ: ۱۳:۴۷ ۱۳۹۲/۱۱/۰۶

من از پروژه سورس باز [LowercaseRoutesMVC](#) که در انتهای مطلب معرفی شد، استفاده می‌کنم. ضمناً این تنظیم، بدون تعریف

[canonical urls](#) ناقص است. یعنی در عمل بر اساس گزارش‌های Google webmaster tools به این نتیجه رسیده‌ام که گوگل، دیتابیس آدرس‌های تکراری را بر اساس تنظیمات canonical urls، به روز و اصلاح می‌کند.

نویسنده: وحید نصیری  
تاریخ: ۱۴:۱۸ ۱۳۹۲/۱۱/۰۶

### نکته‌ای مهم برای کاهش SEO جماعت کپی پیست کار

اگر محتوای سایت شما عینا در سایت‌های دیگر کپی پیست شده، چگونه می‌توانید به گوگل اعلام کنید که محتوای اصلی کدام است؟ کدام یکی کپی است و کدام یک متعلق به شما است؟ برای این منظور باید ذیل مطالب خودتان تنظیمات مربوط به گوگل پلاس خود را نیز [قرار دهید](#) :

```
<a rel="author" href="https://plus.google.com/your_g_plus_id?rel=author">G+</a>
```

سپس در گوگل پلاس خود [اعلام کنید](#) که در این سایت مشارکت دارید (در قسمت Contributor to).

نویسنده: صابر فتح الهی  
تاریخ: ۰:۲۱ ۱۳۹۲/۱۱/۱۹

جالبه که من این فیلتر اعمال کردم

```
<link rel="canonical" href="http://mysite.ir/home/index/5/صفحه-اصلی" />
```

بعد از فراخوانی سایت بدون www سایت ریدایرکت نمیشه اما وقتی کد خروجی سایت بررسی می‌کنیم می‌بینم مثل بالا قسمت canonical اعمال شده، مشکل ممکنه از کجا باشه؟ در حالی که کد درست اجرا میشه اما ریدایرکت رخ نمی‌ده - این بحث چطور بررسی کنم؟

نویسنده: وحید نصیری  
تاریخ: ۰:۳۸ ۱۳۹۲/۱۱/۱۹

- روش دوم اجبار به www با استفاده از ماژول [rewrite](#) در IISهای جدید

```
<system.webServer>
  <rewrite>
    <rules>
      <rule name="Enforce WWW" stopProcessing="true">
        <match url=".*" />
        <conditions>
          <add input="{CACHE_URL}" pattern="^(.+):(?!www)(.*)" />
        </conditions>
        <action type="Redirect" url="{C:1}://www.{C:2}" redirectType="Permanent" />
      </rule>
    </rules>
  </rewrite>
</system.webServer>
```

- روش دوم تولید urlهای lower case

```
<system.webServer>
  <rewrite>
    <rules>
      <rule name="SEO - Lower case" stopProcessing="false">
        <match url="(*)" ignoreCase="false" />
        <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
          <add input="{HTTP_METHOD}" pattern="GET" />
          <add input="{R:1}" pattern="[A-Z]" ignoreCase="false" />
        </conditions>
        <action type="Rewrite" url="{ToLower:{R:1}}" />
      </rule>
    </rules>
```

```
</rewrite>
</system.webServer>
```

ماخذ: [IIS Server through the eyes of an SEO](#)

خلاصه آن:

```
<!-- SEO rules (from: http://www.seomoz.org/blog/what-every-seo-should-know-about-iis#chaining) -->
<!-- SEO | Section 1 | Whitelist -->
<rule name="Whitelist - Resources" stopProcessing="true">
  <match
url="^(?:css/|scripts/|images/|install/|config/|umbraco/|umbraco_client/|base/|webresource\..axd|scriptr
esource\..axd|__browserLink[/^]/arterySignalR/.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
  <action type="None" />
</rule>
<!-- SEO | Section 2 | Rewrites (chaining) -->
<rule name="SEO - Remove default.aspx" stopProcessing="false">
  <match url="(.*)/?default\.aspx$" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
<add input="{HTTP_METHOD}" pattern="GET" />
  </conditions>
  <action type="Rewrite" url="{R:1}" />
</rule>
<rule name="SEO - Remove trailing slash" stopProcessing="false">
  <match url="(.)/$" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
<add input="{HTTP_METHOD}" pattern="GET" />
  <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
  <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
  </conditions>
  <action type="Rewrite" url="{R:1}" />
</rule>
<rule name="SEO - Lower case" stopProcessing="false">
  <match url="(.*)" ignoreCase="false" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
<add input="{HTTP_METHOD}" pattern="GET" />
  <add input="{R:1}" pattern="[A-Z]" ignoreCase="false" />
  </conditions>
  <action type="Rewrite" url="{ToLower:{R:1}}" />
</rule>
<!-- SEO | Section 3 | Redirecting -->
<rule name="SEO - HTTP canonical redirect" stopProcessing="true">
  <match url="^(_*)(.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="true">
<add input="{HTTP_HOST}" pattern="^www\.(.*)" />
  <add input="{HTTP_METHOD}" pattern="GET" />
  <add input="{SERVER_PORT}" pattern="80" />
  </conditions>
  <action type="Redirect" url="http://{C:1}/{R:2}" />
</rule>
<rule name="SEO - HTTPS canonical redirect" stopProcessing="true">
  <match url="^(_*)(.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="true">
<add input="{HTTP_HOST}" pattern="^www\.(.*)" />
  <add input="{HTTP_METHOD}" pattern="GET" />
  <add input="{SERVER_PORT}" pattern="443" />
  </conditions>
  <action type="Redirect" url="https://{C:1}/{R:2}" />
</rule>
<rule name="SEO - Non-canonical redirect" stopProcessing="true">
  <match url="^(_+)(.*)" />
  <conditions logicalGrouping="MatchAll" trackAllCaptures="false">
<add input="{HTTP_METHOD}" pattern="GET" />
  </conditions>
  <action type="Redirect" url="{R:2}" />
</rule>
<!-- // SEO rules -->
```

نویسنده:

وحید نصیری

تاریخ:

۱۳۹۲/۱۱/۲۹ ۱۳:۱۶

یک نکته‌ی تکمیلی در مورد محتوای تکراری

به [گزارش‌های گوگل](#) در مورد سایت جاری که نگاه می‌کردم، تمام لینک‌های به صفحات جستجوی سایت را مثلاً

<http://www.dotnettips.info/search?term=seo> تکراری گزارش کرده بود (تعداد زیادی بودند). چون عنوان قبلی صفحه جستجو، فقط «جستجو» بود (یک عنوان همیشه ثابت). بعد از اینکه عنوان را تبدیل کردم به «جستجو + عبارت وارد شده»، گزارش موارد تکراری آن برطرف شد.

نویسنده: صابر فتح الهی  
تاریخ: ۲۳:۱۸ ۱۳۹۲/۱۲/۰۳

سلام

روش دوم تولید Url های lower case  
در پاسخ شما قسمت تبدیل آدرس به حروف کوچک

```
<action type="Rewrite" url="{ToLower:{R:1}}"/>
```

این روش در تبدیلات مطابق نیاز من درست کار نکرد. به فرض مثال در روش بالا اگر آدرسی مانند [www.mysite.com/Home](http://www.mysite.com/Home) داشته باشیم تبدیل به آدرسی مانند [www.mysite.com/\\_home](http://www.mysite.com/_home) خواهد شد. یک \_ اضافه دارد.

نویسنده: وحید نصیری  
تاریخ: ۰:۲ ۱۳۹۲/۱۲/۰۴

این دستورات چون تعداد زیادی دارند، برای اعمال یکباره آن‌ها، یک \_ را به ابتدای مسیر اضافه می‌کند. در آخر کار این \_ها را در تمام موارد تطابق یافته، حذف کرده و یک redirect را انجام می‌دهد. (این توضیحات [در متن اصلی](#) آن هست)

نویسنده: وحید نصیری  
تاریخ: ۲۳:۲۵ ۱۳۹۳/۰۵/۲۴

اصلاحیه!

امروز در حین گوگل گردی به این آدرس رسیدم (توسط گوگل ایندکس شده بود):  
<http://www.w.thissite.info>  
یعنی وجود [www](http://www.w.thissite.info) در ابتدای آدرس کافی نیست. باید اصل Host دقیقاً بررسی شود و در صورت عدم تطابق، redirect دائمی صورت گیرد.

فیلتر نهایی اصلاح شده: [ForceWww.cs](#)  
نحوه استفاده:

```
filters.Add(new ForceWww("http://www.dotnettips.info/"));
```

نویسنده: محمد زارع  
تاریخ: ۱۸:۳۵ ۱۳۹۳/۱۰/۲۸

سلام. من وقتی از این کلاس ForceWww استفاده میکنم بعد از پابلیش خطای "This webpage has a redirect loop" رو میده. میشه لطفاً راهنمایی کنید اگر نکته خاصی وجود داره

نویسنده: وحید نصیری  
تاریخ: ۱۹:۵ ۱۳۹۳/۱۰/۲۸

- از این فیلتر در حال حاضر در همین سایت جاری استفاده می‌شود. (بنابراین مشکلی ندارد)  
- قسمت new RedirectResult آنرا در سایت خودتان لاگ کنید، تا مشخص شود چندبار و به چه علتی اجرا می‌شود. همچنین بررسی کنید آیا جای دیگری در برنامه Redirect دیگری تنظیم شده یا خیر.