

می دانیم بهینه سازی موتورهای جستجو (به انگلیسی: Search engine optimization (SEO)) که گاهی در فارسی به آن سئو نیز گفته می شود، عملیاتی است برای بهبود دید یک وب گاه یا یک صفحه وب، در صفحه نتایج موتورهای جستجو که می تواند طبیعی و یا الگوریتمی باشد. این عملیات برای وبمسترها یکی از عوامل مهم و حیاتی بدست آوردن کاربران جدید از موتورهای جستجو است.

اگر [چک لیست های SEO وب سایت ها](#) را مشاهده کنیم، می توانیم آن ها را در دو دسته ی کلی [بهینه سازی درونی و برونی وب سایت](#) در نظر بگیریم:

Off-Page Optimization یا برونی، که بیشتر بر دوش مشاوران سئو و خود مدیران وب سایت است. (link building، فعالیت در شبکه اجتماعی و ...) و اما در حوزه **On-Page Optimization** یا درونی که بخش های مهمی از آن وظیفه ی مابرنامه نویسی ها است. (H1 Tag، URL Naming، Meta Tags، عنوان صفحه و ...) [البته عامل درونی بهینه سازی محتوا (Content Optimization) که مهمترین عامل در الگوریتم های نسل جدید موتورهای جستجو و همچنین [الگوریتم جدید گوگل](#) (و +) به حساب می آید بر عهده مشاوران سئو و خود مدیران وب سایت می باشد]

در ادامه به ارائه چند راهکار جهت بهینه سازی برنامه های وب ASP.NET مان برای موتورهای جستجو می پردازیم:

1. متدی برای ایجاد عنوان سایت

```
private const string SeparatorTitle = " - ";
private const int MaxLenghtTitle = 60;
public static string GeneratePageTitle(params string[] crumbs)
{
    var title = "";

    for (int i = 0; i < crumbs.Length; i++)
    {
        title += string.Format(
            "{0}{1}",
            crumbs[i],
            (i < crumbs.Length - 1) ? SeparatorTitle : string.Empty
        );
    }

    title = title.Substring(0, title.Length <= MaxLenghtTitle ? title.Length :
MaxLenghtTitle).Trim();

    return title;
}
```

نکته:

MaxLenghtTitle پیشنهادی برای عنوان سایت 60 می باشد.

2. متدی برای ایجاد متاتگ صفحات سایت

```
public enum CacheControlType
{
    [Description("public")]
    public,
    [Description("private")]
    private,
    [Description("no-cache")]
    nocache,
    [Description("no-store")]
    nostore
}
```

```
}

private const int MaxLenghtTitle = 60;
private const int MaxLenghtDescription = 170;
private const string FaviconPath = "~/cdn/ui/favicon.ico";
public static string GenerateMetaTag(string title, string description, bool allowIndexPage, bool
allowFollowLinks, string author = "", string lastmodified = "", string expires = "never", string
language = "fa", CacheControlType cacheControlType = CacheControlType._private)
{
    title = title.Substring(0, title.Length <= MaxLenghtTitle ? title.Length :
MaxLenghtTitle).Trim();
    description = description.Substring(0, description.Length <= MaxLenghtDescription ?
description.Length : MaxLenghtDescription).Trim();

    var meta = "";
    meta += string.Format("<title>{0}</title>\n", title);
    meta += string.Format("<link rel=\"shortcut icon\" href=\"{0}\"/>\n", FaviconPath);
    meta += string.Format("<meta http-equiv=\"content-language\" content=\"{0}\"/>\n", language);
    meta += string.Format("<meta http-equiv=\"content-type\" content=\"text/html; charset=utf-
8\"/>\n");
    meta += string.Format("<meta charset=\"utf-8\"/>\n");
    meta += string.Format("<meta name=\"description\" content=\"{0}\"/>\n", description);
    meta += string.Format("<meta http-equiv=\"Cache-control\" content=\"{0}\"/>\n",
EnumExtensions.EnumHelper<CacheControlType>.GetEnumDescription(cacheControlType.ToString()));
    meta += string.Format("<meta name=\"robots\" content=\"{0}, {1}\" />\n", allowIndexPage ?
"index" : "noindex", allowFollowLinks ? "follow" : "nofollow");
    meta += string.Format("<meta name=\"expires\" content=\"{0}\"/>\n", expires);

    if (!string.IsNullOrEmpty(lastmodified))
        meta += string.Format("<meta name=\"last-modified\" content=\"{0}\"/>\n", lastmodified);

    if (!string.IsNullOrEmpty(author))
        meta += string.Format("<meta name=\"author\" content=\"{0}\"/>\n", author);

    //-----Google & Bing Doesn't Use Meta Keywords ...
    //meta += string.Format("<meta name=\"keywords\" content=\"{0}\"/>\n", keywords);

    return meta;
}
```

چند نکته :

MaxLenghtDescription پیشنهادی برای متاتگ توضیح سایت 170 می باشد.

آشنایی با متاتگ‌ها (Meta tags) و کاربرد آنها در صفحات وب (HTML)

برای کاربرد allowIndexPage و allowFollowLinks هم می‌توانید به لینک بالا و بررسی متاتگ robots بپردازید.

با توجه به اهمیت شبکه‌های اجتماعی متاتگ‌های شبکه‌های اجتماعی (+ و +) را هم نباید از قلم انداخت.

برای دریافت Description نوع سفارشی CacheControlType از پروژه متدهای الحاقی علیرضا اسم رام استفاده کردم.

3.متدی برای ایجاد Slug (اسلاگ آدرسی با مفهوم برای بکار بردن در URL ها است که دوست‌دار موتورهای جستجو می‌باشد)

```
private const int MaxLenghtSlug = 45;
public static string GenerateSlug(string title)
{
    var slug = RemoveAccent(title).ToLower();
    slug = Regex.Replace(slug, @"[a-z0-9-\u0600-\u06FF]", "-");
    slug = Regex.Replace(slug, @"\s+", "-").Trim();
    slug = Regex.Replace(slug, @"-+", "-");
    slug = slug.Substring(0, slug.Length <= MaxLenghtSlug ? slug.Length : MaxLenghtSlug).Trim();

    return slug;
}

private static string RemoveAccent(string text)
{
    var bytes = Encoding.GetEncoding("UTF-8").GetBytes(text);
    return Encoding.UTF8.GetString(bytes);
}
```

نکته :

MaxLengthSlug پیشنهادی برای عنوان سایت 45 می‌باشد.

نمونه ای از کاربرد توابع :

```
Head.InnerHtml = SEO.GenerateMetaTag
(
    title: SEO.GeneratePageTitle(".NET Tips", "آرشیو مطالب", "ASP.NET MVC
#1"),
    description: "ASP.NET MVC با وجود فریم ورک پخته‌ای به نام ASP.NET MVC چرا
forms, مطرح می‌شود این است: «برای چی؟». بنابراین تا به این سؤال ASP.NET MVC اولین سؤالی که
پاسخ داده نشود, هر نوع بحث فنی در این مورد بی فایده است",
    allowIndexPage: true,
    allowFollowLinks: true,
    author: "وحید نصیری",
    cacheControlType: SEO.CacheControlType._private
);
```

و خروجی در Page Source :

```
<title>.NET Tips - آرشیو مطالب - ASP.NET MVC #1</title>
<link rel="shortcut icon" href="../../cdn/images/ui/favicon.ico"/>
<meta http-equiv="content-language" content="fa"/>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<meta charset="utf-8"/>
<meta name="description" content="ASP.NET MVC با وجود فریم ورک پخته‌ای به نام ASP.NET MVC چرا
..."/>
<meta http-equiv="Cache-control" content="private"/>
<meta name="robots" content="index, follow" />
<meta name="expires" content="never"/>
<meta name="author" content="وحید نصیری"/>
```

موفق باشید

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۵:۲۰ ۱۳۹۲/۰۹/۰۸

دوست من با این روش

```
Head.InnerHtml = SEO.GenerateMetaTag(...)
```

در صورتی که در قسمت head صفحه استایل یا متاتگهای دیگری تعریف شده باشد همگی حذف خواهند شد.

نویسنده: محسن نجف زاده
تاریخ: ۱۸:۴۵ ۱۳۹۲/۰۹/۰۸

بله صابر جان حتما همین طور است (برای مثال بیان شده است).
خودم به این صورت استفاده می‌کنم:

```
<head>  
  <asp:Literal runat="server" ID="litHead"></asp:Literal>  
  
  ....  
  
</head>
```

و در قسمت *code - behind*

```
litHead.Text = SEO.GenerateMetaTag(...)
```

پ ن : چرا از کنترل Literal استفاده شده و از Label استفاده نشده ؟ به این دلیل که خروجی متن رندر شده Label توسط تگ `` یا `<label>` احاطه می‌شود. ([+](#))

نویسنده: پوریا منفرد
تاریخ: ۱۲:۱۵ ۱۳۹۲/۱۰/۱۴

با سلام

امکان داره این خط کلاسش رو هم بزارید بنده دقیقا نتونستم با متد الحاقی که گفتین این قسمت رو پیاده سازی کنم

```
EnumExtensions.EnumHelper<CacheControlType>.GetEnumDescription(cacheControlType.ToString());
```

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۳ ۱۳۹۲/۱۰/۱۴

مراجعه کنید به پروژه [DNT.Extensions](#) و همچنین [اینجا](#)

نویسنده: پوریا منفرد
تاریخ: ۱۳:۲۸ ۱۳۹۲/۱۰/۱۴

بنده یه مشکلی دارم : از MasterPage استفاده می‌کنم در ASP.net Webform و فرمی رو بهش Connect کردم که:

1- در MPage یک سری متا تگ‌ها هست CSS + JavaScript

2- وقتی در یک webForm متصل به Mpage کد:

```
Head.InnerHtml = SEO.GenerateMetaTag ....
```

رو قرار می‌دهم تمام تگ‌های Head حتی MasterPage هم پاک می‌شه
اینم می‌دونم این کلاس این کارو می‌کنه
راه حل چیه که تگ‌ها پاک نشه ؟ مخصوصا آدرس‌های CSS و JavaScript ؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۱ ۱۳۹۲/۱۰/۱۴

[کمی بالاتر توضیح دادند](#) ؛ استفاده از یک Literal برای مثال.

نویسنده: مهرداد پاک دل
تاریخ: ۱۶:۱۶ ۱۳۹۲/۱۱/۰۲

من این توابع را خواندم که در مورد seo نوشته بودید ولی متاسفانه نحوه استفاده از آن‌ها را نفهمیدم یعنی کجا تعریف میشن و از کجا پارامترهای خود را دریافت می‌کنند. لطف کنید اگر امکان داره یه پروژه کوچک همه این‌ها استفاده شده باشه اگر هست بگذارید .
نکته دیگه شما عنوان هر پست را داخل url‌ها نمایش میدید منظور اینکه جزئی از url سایت است این کار را چگونه انجام می‌دهید

```
mysite.info/post/1
```

```
dotnettips.info/post/1200/seo
```

نویسنده: محسن خان
تاریخ: ۱۶:۳۸ ۱۳۹۲/۱۱/۰۲

- [پروژه Iris](#) هست. به section metatags اون دقت کن در فایل‌های View.

- مثلا متد Head.InnerHtml = SEO.GenerateMetaTag عنوان شده در این مطلب باید در Page_Load یک وب فرم فراخوانی شود. Id مطلب رو دارید. عنوان و متن و سایر مشخصات اون رو از دیتابیس دریافت کنید و بعد فقط یک جایگذاری است در متد تهیه شده.

- به اینکار Routing و Url rewrite می‌گن. بحثش [در MVC](#) و [وب فرم‌ها](#) کمی با هم فرق می‌کنه.

نویسنده: امین قادری
تاریخ: ۳:۳۵ ۱۳۹۳/۰۵/۰۹

لطفا مثالی را در مورد نحوه‌ی استفاده از " **متدی برای ایجاد Slug** " در عمل ارسال نمائید. با تشکر