عنوان: **حذف اعراب از حروف و کلمات**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۰ ۰:۱۰
آدرس: www.dotnettips.info
گروه‌ها: Security, Persian, Unicode

برای بهبود قسمت ثبت نام در یک سایت بهتر است بین «وحید» و «وَحید» تفاوتی قائل نشد. این مورد ممکن است خصوصا حین ارسال پیام‌های خصوصی در آینده جهت تشخیص افراد مشکل ساز شود. همچنین در تهیه slug برای نمایش در Urlها نیز باید اعراب را حذف کرد. منظور از slug، عنوان کوتاهی است که در انتهای یک آدرس ممکن است ذکر شود.

```
http://www.site.com/post/12/slug
```

سؤال: چگونه می‌توان اعراب را از متون فارسی یا عربی حذف کرد؟

متد انجام اینکار را در ذیل مشاهده می‌کنید:

```
using System.Globalization;
using System.Text;

    static string RemoveDiacritics(string text)
    {
        var normalizedString = text.Normalize(NormalizationForm.FormD);
        var stringBuilder = new StringBuilder();

        foreach (var c in normalizedString)
        {
            var unicodeCategory = CharUnicodeInfo.GetUnicodeCategory(c);
            if (unicodeCategory != UnicodeCategory.NonSpacingMark)
            {
                stringBuilder.Append(c);
            }
        }

        return stringBuilder.ToString().Normalize(NormalizationForm.FormC);
    }
```

توضیحات

متد Normalize با پارامتر NormalizationForm.FormD، سبب می‌شود تا کاراکترها به گلیف‌های اصلی تشکیل دهنده‌ی آن‌ها تجزیه شوند. به عبارتی، حروف از اعراب جدا خواهند شد. در ادامه این کاراکترها اسکن شده و صرفا مواردی که حروف پایه را تشکیل می‌دهند، جمع آوری و بازگشت داده می‌شوند. حالت NormalizationForm.FormC که در انتها بکار گرفته شده، برعکس است.
در یونیکد یک حرف می‌تواند از یک یا چند code point تشکیل شود. در حالت FormC، هر حرف با اعراب آن یک code point را تشکیل می‌دهند. در حالت FormD، حرف با اعراب آن دو code point را تشکیل خواهند داد. به همین جهت در ابتدای کار، رشته تبدیل به حالت D شده تا بتوان اعراب آن‌را مجزای از حروف پایه حذف کرد.
البته اعراب در اینجا به اعراب عربی ختم نمی‌شود. یک سری حروف اروپایی مانند "ö"، "ä" و "ü" را نیز شامل می‌شود.

نظرات خوانندگان

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۳/۰۲/۱۱ ۱۶:۱۲

اطلاعات بیشتر در این پرسش و پاسخ .
لیست کاراکترهای یونیکد از نوع NonSpacingMark

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۳/۰۲/۱۱ ۱۶:۴۴

یک سوال: علت استفاده از حالت FormC در انتهای کد چیست؟ چرا فقط به کد زیر بسنده نکردیم:

```
return stringBuilder.ToString();
```

بوسیله Normalize، می‌توانیم خروجی را با مقدار string دیگر مقایسه نماییم یا بعبارت دیگر خروجی مقایسه پذیر خواهد شد. در این پرسش و پاسخ بیشتر درباره Normalize بحث شده است.

نویسنده: داوود
تاریخ: ۱۳۹۳/۰۲/۱۳ ۸:۱۳

با سلام
آیا تنوین و تشدید در این حالت جز اعراب محسوب میشوند
و همچنین ی (یای عربی) جز حروف اعراب دار است
تشکر بابت مطلب مفیدتون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۳ ۹:۲

- بله.
- خیر.

نویسنده: علیرضا
تاریخ: ۱۳۹۳/۰۲/۱۳ ۱۴:۳۹

با سلام. برای سرچ یک کلمه بدون اعراب در متنی پر از اعراب باید به چه صورت عمل کرد که بهینه باشد؟
مثلا کلمه‌ی محمد را بخواهیم در دیتابیسی که متن کل قرآن است سرچ کنیم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۳ ۱۴:۵۶

جستجوی بهینه‌ی متنی بر روی حجم بالایی از اطلاعات بهتر است توسط روش‌های full text search انجام شود. مثلا از لوسین استفاده کنید، به همراه Lucene.Net.Analysis.Analyzer.ArabicAnalyzer آن که مخصوص جستجو بر روی متون عربی است.
همچنین اگر از FTS در SQL Server استفاده می‌کنید باید از accent insensitive collate استفاده کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۴ ۲۳:۱۹

اصلاحیه!

کدهای فوق «آ» را تبدیل به «ا» می‌کنند. مشکلی بود که در حین ثبت نام پیش آمده بود. «آفتاب» برای مثال تبدیل به «افتاب»

می‌شد. برای رفع، داخل حلقه:

```
if (unicodeCategory != UnicodeCategory.NonSpacingMark)
                {
                    stringBuilder.Append(c);
                }
                else
                {
                    //اسامی مانند آفتاب نباید خراب شوند
                    if (c == 1619) //آ
                    {
                        stringBuilder.Append(c);
                    }
                }
```