عنوان: **تبدیل html به pdf با کیفیت بالا**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۰ ۹:۵۴
آدرس: www.dotnettips.info
گروه‌ها: iTextSharp, wkhtmltopdf, WkHtmlToXSharp

کتابخانه iTextSharp دارای کلاسی است به نام HTMLWorker که کار تبدیل عناصر HTML را به عناصر متناظر خودش، انجام می‌دهد. این کلاس در حال حاضر منسوخ شده درنظر گرفته می‌شود (اینطور توسط نویسندگان آن اعلام شده) و دیگر توسعه نخواهد یافت. بنابراین اگر از HTMLWorker استفاده می‌کنید با یک کلاس قدیمی که دارای HTML Parser ایی بسیار بدوی است طرف هستید و در کل برای تبدیل محتوای HTML ایی با ساختار بسیار ساده بد نیست؛ اما انتظار زیادی از آن نداشته باشید.
جایگزین کلاس HTMLWorker در این کتابخانه در حال حاضر کتابخانه itextsharp.xmlworker است، که به صورت یک افزونه در کنار کتابخانه اصلی در حال توسعه می‌باشد. مشکل اصلی این کتابخانه، عدم پشتیبانی از UTF8 و راست به چپ است. بنابراین حداقل به درد کار ما نمی‌خورد.

راه حل بسیار بهتری برای موضوع اصلی بحث ما وجود دارد و آن هم استفاده از موتور WebKit (همان موتوری که برای مثال در Apple Safari استفاده می‌شود) برای HTML parsing و سپس تبدیل نتیجه نهایی به PDF است. پروژه‌ای که این مقصود را میسر کرده، wkhtmltopdf نام دارد.
توسط آن به کمک موتور WebKit، کار HTML Parsing انجام شده و سپس برای تبدیل عناصر نهایی به PDF از امکانات کتابخانه‌ای به نام QT استفاده می‌شود. کیفیت نهایی آن کپی مطابق اصل HTML قابل مشاهده در یک مرورگر است و با یونیکد و زبان فارسی هم مشکلی ندارد.

برای استفاده از این کتابخانه‌ی native در دات نت، شخصی پروژه‌ای را ایجاد کرده است به نام WkHtmlToXSharp که محصور کننده‌ی wkhtmltopdf می‌باشد. در ادامه به نحوه استفاده از آن خواهیم پرداخت:

الف) دریافت پروژه WkHtmlToXSharp
پروژه WkHtmlToXSharp را از آدرس زیر می‌توانید دریافت کنید.
https://github.com/pruiz/WkHtmlToXSharp

این پروژه به همراه فایل‌های کامپایل شده نهایی wkhtmltopdf نیز می‌باشد و حجمی حدود 40 مگ دارد. به علاوه فعلا نسخه 32 بیتی آن در دسترس است. بنابراین باید دقت داشت که نباید تنظیمات پروژه دات نت خود را بر روی Any CPU قرار دهیم، زیرا در این حالت برنامه شما در یک سیستم 64 بیتی بلافاصله کرش خواهد کرد. تنظیمات target platform پروژه دات نتی ما حتما باید بر روی X86 تنظیم شود.

ب) پس از دریافت این پروژه و افزودن ارجاعی به اسمبلی WkHtmlToXSharp.dll، استفاده از آن به نحو زیر می‌باشد:

```
using System.IO;
using WkHtmlToXSharp;
using System;

namespace Test2
{
    public class WkHtmlToXSharpTest
    {
        public static void ConvertHtmlStringToPdfTest()
        {
            using (var wk = new MultiplexingConverter())
            {
                wk.Begin += (s, e) => Console.WriteLine("Conversion begin, phase count: {0}", e.Value);
                wk.Error += (s, e) => Console.WriteLine(e.Value);
                wk.Warning += (s, e) => Console.WriteLine(e.Value);
                wk.PhaseChanged += (s, e) => Console.WriteLine("PhaseChanged: {0} - {1}", e.Value,
e.Value2);
                wk.ProgressChanged += (s, e) => Console.WriteLine("ProgressChanged: {0} - {1}",
e.Value, e.Value2);
```

```
                wk.Finished += (s, e) => Console.WriteLine("Finished: {0}", e.Value ? "success" :
"failed!");

                wk.GlobalSettings.Margin.Top = "0cm";
                wk.GlobalSettings.Margin.Bottom = "0cm";
                wk.GlobalSettings.Margin.Left = "0cm";
                wk.GlobalSettings.Margin.Right = "0cm";

                wk.ObjectSettings.Web.EnablePlugins = false;
                wk.ObjectSettings.Web.EnableJavascript = false;
                wk.ObjectSettings.Load.Proxy = "none";

                var htmlString = File.ReadAllText(@"c:\page.xhtml");
                var tmp = wk.Convert(htmlString);

                File.WriteAllBytes(@"tst.pdf", tmp);
            }
        }
    }
}
```

کار با وهله سازی از کلاس MultiplexingConverter شروع می‌شود. اگر علاقمند باشید که درصد پیشرفت کار به همراه خطاهای احتمالی پردازشی را ملاحظه کنید می‌توان از رخدادگردان‌هایی مانند ProgressChanged و Error استفاده نمائید که نمونه‌ای از آن در کد فوق بکارگرفته شده است.

تبدیل HTML به PDF آنچنان تنظیمات خاصی ندارد زیرا فرض بر این است که قرار است از همان تنظیمات اصلی HTML مورد نظر استفاده گردد. اما اگر نیاز به تنظیمات بیشتری وجود داشت، برای مثال به کمک GlobalSettings آن می‌توان حاشیه‌های صفحات فایل نهایی تولیدی را تنظیم کرد.

موتور WebKit با توجه به اینکه موتور یک مرورگر است، امکان پردازش جاوا اسکریپت را هم دارد. بنابراین اگر قصد استفاده از آن‌را نداشتید می‌توان خاصیت ObjectSettings.Web.EnableJavascript را به false مقدار دهی کرد.

کار اصلی، در متد Convert انجام می‌شود. در اینجا می‌توان یک رشته را که حاوی فایل HTML مورد نظر است به آن ارسال کرد و نتیجه نهایی، آرایه‌ای از بایت‌ها، حاوی فایل باینری PDF تولیدی است.

روش دیگر استفاده از این کتابخانه، مقدار دهی wk.ObjectSettings.Page می‌باشد. در اینجا می‌توان Url یک صفحه اینترنتی را مشخص ساخت. در این حالت دیگر نیازی نیست تا به متد Convert پارامتری را ارسال کرد. می‌توان از overload بدون پارامتر آن استفاده نمود.

یک نکته:

اگر می‌خواهید زبان فارسی را توسط این کتابخانه به درستی پردازش کنید، نیاز است حتما یک سطر زیر را به header فایل html خود اضافه نمائید:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

نظرات خوانندگان

نویسنده: حمید
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۳:۴۲

سلام.خسته نباشید.ممنون از لطلاعات پربارتون.
بنده با توجه به توضیحاتتوون عمل کردم و خروجی Pdf هم دریافت کردم
اما مشکلی که وجود داره،بیشتر وقتها ارور میده و ویژوال استودیو بسته می‌شه و میگه فضای کافی برای لود این dll وجود نداره.(WkHtmlToXSharp)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۴:۱۵

در عمل عموم کدهای native نوشته شده با سی پلاس پلاس این مشکلات را دارند:
- ناپایدار
- دارای نشتی‌های حافظه بالا
- نا امن
- نیاز به کامپایل مجزا برای سیستم‌های 64 بیتی و 32 بیتی
فقط از این کلمه لذت می‌برند: «سرعت»! اما در 4 مورد فوق حرفی برای گفتن ندارند.

ولی خوب بازسازی این پروژه‌ها با دات نت وقت زیادی می‌گیرد به همین جهت کسی طرف تبدیل آن‌ها نرفته. نوشتن یک html parser خوب و تمام عیار، یک پروژه چند میلیون دلاری است که موزیلا، مایکروسافت، اپل، گوگل و غیره درگیر آن هستند!

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۵/۲۲ ۱۹:۴۸

با سلام؛
ببخشید شما برای قابلیت نسخه چاپی این سایت از QT استفاده کرده اید یا iTextSharp؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۲ ۲۰:۱۸

ترکیب iTextSharp و Html Agility Pack است. به عبارتی مجبور شدم قابلیت تبدیل html به pdf غیرمطلوب iTextSharp رو در حد نیاز سایت از صفر بازنویسی کنم. html parser سورس باز Html Agility Pack واقعا باکیفیت است.

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۵/۲۲ ۲۰:۵۳

من می‌خواهم در یک وب سایت که یک اپلیکیشن است یک سری گزارش تهیه بکنم سناریو به این صورت است که ابتدا درون وب سایت گزارش را نمایش می‌دهیم و اگر کاربر خواست می‌تواند اون گزارش رو به فرمت PDF دانلود کند. برای این سناریو به نظر شما از چه چیزی استفاده کنم؟

ممنونم

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۵/۲۲ ۲۱:۱۵

باید برنامه نویسی کنید. تبدیل html به pdf در این نوع موارد خاص جواب نمی‌دهد چون یک گزارش نیاز به خیلی از جزئیات دیگر مانند شماره صفحه، header و footer و غیره هم دارد. از این مثال می‌تونید ایده بگیرید.

نویسنده: hasani
تاریخ: ۱۳۹۱/۰۵/۲۳ ۲۳:۲۵

شاید این مثال هم کمک کند http://www.codeproject.com/Articles/260470/PDF-reporting-using-ASP-NET-MVC3

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۴ ۰:۲۷

از این روش استفاده می‌کنه. به علاوه باید درنظر داشت HTMLWorker کتابخانه iTextSharp منسوخ شده در نظر گرفته می‌شود و دیگر توسعه نخواهد یافت و حتی نگهداری هم نمی‌شود. با Xml Worker آن جایگزین شده که فعلا از RTL و یونیکد پشتیبانی نمی‌کند.

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۵/۲۹ ۲۰:۰

ببخشید استاد می‌خواست بپرسم که وقتی که در وب سایت مطالب رو از نسخه چاپی دانلود می‌کنیم نام تمام فایل‌ها dotnettips هست آیا شما یه همچین فایل فیزیکی دارید که محتوای آن را به صورت داینامیک ایجاد می‌کنید یا اینکه درون حافظه این فایل را ایجاد می‌کنید و اصلاً فایل فیزیکی وجود ندارد؟ اگه از حالت فایل فیزیکی استفاده می‌کنید تداخل داده به وجود نمی‌آید؟ مثلاً یکی از کاربران روی مقاله 100 کلیک کند و هم زمان یک کاربر دیگر روی مقاله 101 کلیک کند اون موقعه متن مقاله 101 برای هر 2 نمایش داده می‌شود

ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۹ ۲۰:۱۳

iTextSharp با Stream کار می‌کند. این استریم می‌تواند فایل استریم یا memory stream باشد؛ که من در اینجا از حالت memory stream استفاده می‌کنم.
ضمن اینکه اگر فایل استریم هم می‌بود چون نام فایل‌ها یکی نیست، تداخلی رخ نمی‌داد.

نویسنده: نوید
تاریخ: ۱۳۹۱/۰۸/۰۶ ۱۴:۴۴

جناب نصیری سلام
از مقالات آموزنده شما بسیار سپاسگذارم
میخاستم بدونم شما در سایتتون از iTextSharp استفاده کردید یا PdfReport ؟
چون من از iTextSharp استفاده کردم و داخل جداولم متون فارسی نوشتم که کلا به هم ریخته نمایش میده ، اگه امکانش هست یک راهنمایی بفرمائید
متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۶ ۱۶:۱۱

لطفا برچسب iTextSharp را در سایت جاری دنبال بفرمائید. در این مورد کاملا توضیح داده شده. برای نمونه: (^)

نویسنده: سامان

تاریخ: ۱۳۹۲/۰۲/۲۷ ۲۱:۳۰

سلام
پروژه wkhtmltopdf برای ویندوز فقط معماری i386 رو پشتیبانی می‌کنه. آیا این میتونه برای یه برنامه که قراره رو سیستم مشتری‌های مختلفی اجرا بشه مشکل ایجاد کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۷ ۲۲:۹

بله. platform target رو باید روی X86 قرار بدید تا همه جا بدون مشکل اجرا بشه.
یا اینکه نسخه 64 بیتی اون رو هم پیدا یا کامپایل کنید و نهایتا مانند خیلی از کارهای native باید دو نسخه 64 بیتی و 32 بیتی از برنامه خودتون رو منتشر کنید.

نویسنده: آتوسا فتوحی
تاریخ: ۱۳۹۳/۰۳/۱۱ ۱۰:۷

اگر از ابزارهای گزارش گیری مانند: Telerik, Stimul, ... استفاده شود, بصورت Built-in قابلیت تبدیل به PDF و یا Excel, ... را به ما می‌دهد.