عنوان: **تبدیل تعدادی تصویر به یک فایل PDF**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۴/۰۸ ۲۳:۵۳:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: iTextSharp

صورت مساله:
تعدادی تصویر داریم، می‌خواهیم این‌ها را تبدیل به فایل PDF کنیم به این شرط که هر تصویر در یک صفحه مجزا قرار داده شود.
در ادامه برای این منظور از کتابخانه‌ی iTextSharp استفاده خواهیم کرد.

iTextSharp چیست؟
iTextSharp کتابخانه‌ی سورس باز و معروفی جهت تولید فایل‌های PDF ، توسط برنامه‌های مبتنی بر دات نت است. آن را از آدرس زیر می‌توان دریافت کرد:

/http://sourceforge.net/projects/itextsharp

کتابخانه iTextSharp نیز جزو کتابخانه‌هایی است که از جاوا به دات نت تبدیل شده‌اند. نام کتابخانه اصلی iText است و اگر کمی جستجو کنید می‌توانید کتاب iText in Action به صفحه‌ای 617 از انتشارات MANNING را در این مورد نیز بیابید. هر چند این کتاب برای برنامه نویس‌های جاوا نوشته شده اما نام کلاس‌ها و متدها در iTextSharp تفاوتی با iText اصلی ندارند و مطالب آن برای برنامه نویس‌های دات نت هم قابل استفاده است.

مجوز استفاده از iTextSharp کدام است؟
مجوز این کتابخانه GNU Affero General Public License است. به این معنا که شما موظفید، تغییری در قسمت تهیه کننده خواص فایل PDF تولیدی که به صورت خودکار به نام کتابخانه تنظیم می‌شود، ندهید. اگر می‌خواهید این قسمت را تغییر دهید باید هزینه کنید. همچنین با توجه به اینکه این مجوز، GPL است یعنی زمانیکه از آن استفاده کردید باید کار خود را به صورت سورس باز ارائه دهید؛ درست خوندید! بله! مثل مجوز استفاده از نگارش عمومی و رایگان MySQL و اگر نمی‌خواهید اینکار را انجام دهید، در اینجا تاکید شده که باید کتابخانه را خریداری کنید.

نحوه استفاده از کتابخانه iTextSharp
در ابتدا کد تبدیل تصاویر به فایل PDF را در ذیل مشاهده خواهید کرد. فرض بر این است که ارجاعی را به اسمبلی itextsharp.dll اضافه کرده‌اید:

```
using System.Collections.Generic;
using System.Drawing.Imaging;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;

namespace iTextSharpTests
{
   public class ImageToPdf
   {
       public iTextSharp.text.Rectangle PdfPageSize { set; get; }
       public ImageFormat ImageCompressionFormat { set; get; }
       public bool FitImagesToPage { set; get; }

       public void ExportToPdf(IList<string> imageFilesPath, string outPdfPath)
       {
           using (var pdfDoc = new Document(PdfPageSize))
           {
               PdfWriter.GetInstance(pdfDoc, new FileStream(outPdfPath, FileMode.Create));
               pdfDoc.Open();

               foreach (var file in imageFilesPath)
               {
                   var pngImg = iTextSharp.text.Image.GetInstance(file);
```

```
                    if (FitImagesToPage)
                    {
                        pngImg.ScaleAbsolute(pdfDoc.PageSize.Width, pdfDoc.PageSize.Height);
                    }
                    pngImg.SetAbsolutePosition(0, 0);

                    //add to page
                    pdfDoc.Add(pngImg);
                    //start a new page
                    pdfDoc.NewPage();
                }
            }
        }
    }
}
```

توضیحات:

استفاده از کتابخانه‌ی iTextSharp همیشه شامل 5 مرحله است. ابتدا شیء Document ایجاد می‌شود. سپس وهله‌ای از PdfWriter ساخته شده و Document جهت نوشتن در آن گشوده خواهد شد. در طی یک سری مرحله محتویات مورد نظر به Document اضافه شده و نهایتا این شیء بسته خواهد شد. البته در اینجا چون کلاس Document اینترفیس IDisposable را پیاده سازی کرده، بهترین روش استفاده از آن بکارگیری واژه کلیدی using جهت مدیریت منابع آن است. به این ترتیب کامپایلر به صورت خودکار قطعه try/finally مرتبط را جهت پاکسازی منابع، تشکیل خواهد داد.

اندازه صفحات توسط سازنده‌ی شیء Document مشخص خواهند شد. این شیء از نوع iTextSharp.text.Rectangle است؛ اما مقدار دهی آن توسط کلاس iTextSharp.text.PageSize صورت می‌گیرد که انواع و اقسام اندازه صفحات استاندارد در آن تعریف شده‌اند.

متد iTextSharp.text.Image.GetInstance که در این مثال جهت دریافت اطلاعات تصاویر مورد استفاده قرار گرفت، 15 overload دارد که از آدرس مستقیم یک فایل تا استریم مربوطه تا Uri یک آدرس وب را نیز می‌پذیرد و از این لحاظ بسیار غنی است.

مثالی در مورد نحوه استفاده از کلاس فوق:

```
using System.Collections.Generic;
using System.Drawing.Imaging;

namespace iTextSharpTests
{
   class Program
   {
       static void Main(string[] args)
       {
           new ImageToPdf
           {
               FitImagesToPage = true,
               ImageCompressionFormat = ImageFormat.Jpeg,
               PdfPageSize = iTextSharp.text.PageSize.A4
           }.ExportToPdf(
                   imageFilesPath: new List<string>
                           {
                               @"D:\3.jpg",
                               @"D:\4.jpg"
                           },
                   outPdfPath: @"D:\tst.pdf"
                   );
       }
   }
}
```

نظرات خوانندگان

نویسنده: Nima
تاریخ: ۱۳۹۰/۰۶/۱۲ ۲۲:۲۱:۲۵

سلام آقای نصیری
ممنون از آموزش مفیدتون. من این سمپل رو برای خودم نوشتم و فقط FitImagesToPage = false قرار دادم اما عکسها به گوشه پایین سمت چپ میچسبه. میخواستم بدونم چکار باید بکنم که عکس در وسط صفحه نمایش داده بشه؟
باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۱۲ ۲۲:۳۳:۵۹

باید همان متد SetAbsolutePosition را مقدار دهی کنید. مثلا به این صورت:

```
pngImg.SetAbsolutePosition((pdfDoc.PageSize.Width - pngImg.Width) / 2, (pdfDoc.PageSize.Height - pngImg.Height) / (2
```

نویسنده: Nima
تاریخ: ۱۳۹۰/۰۶/۱۲ ۲۲:۵۸:۵۷

بسیار عالی بود با تشکر

نویسنده: ali
تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۶:۴۸

با سلام من از این روش استفاده کردم فقط مشکلی که وجود داشت عکسها را از عرض میکشه علت از چی میتونه باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۷:۱۱

مربوط به گزینه FitImagesToPage است.