

ابتدا مثال ساده زیر را در نظر بگیرید:

```
using System.Diagnostics;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;

namespace OptimizeImageSizes
{
    class Program
    {
        static void Main(string[] args)
        {
            test1();
            test2();
        }

        private static void test2()
        {
            using (var pdfDoc = new Document(PageSize.A4))
            {
                var pdfWriter = PdfWriter.GetInstance(pdfDoc, new FileStream("Test2.pdf",
                    FileMode.Create));
                pdfDoc.Open();

                var table = new PdfPTable(new float[] { 1, 2 });
                table.AddCell(Image.GetInstance("myImage.png"));
                table.AddCell(Image.GetInstance("myImage.png"));
                pdfDoc.Add(table);
            }

            Process.Start("test2.pdf");
        }

        private static void test1()
        {
            using (var pdfDoc = new Document(PageSize.A4))
            {
                var pdfWriter = PdfWriter.GetInstance(pdfDoc, new FileStream("Test1.pdf",
                    FileMode.Create));
                pdfDoc.Open();

                var table = new PdfPTable(new float[] { 1, 2 });
                var image = Image.GetInstance("myImage.png");
                table.AddCell(image);
                table.AddCell(image);
                pdfDoc.Add(table);
            }

            Process.Start("test1.pdf");
        }
    }
}
```

در اینجا یک تصویر به نام myImage.png به دو طریق، به صفحه‌ای اضافه شده است:

الف) در متد test1، یک وهله از آن تهیه و دو بار به صفحه اضافه شده است.

ب) در متد test2، به نحوی متداول، هربار که نیاز به نمایش تصویری بوده، یک وهله جدید از تصویر تهیه و اضافه شده است.

نکته‌ی مهم در اینجا، حجم نهایی دو فایل حاصل است:

حجم فایل test2.pdf دقیقاً دو برابر حجم فایل test1.pdf است. علت هم به این بر می‌گردد که هر وهله جدیدی از شیء Image،

صرفنظر از محتوای آن، توسط iTextSharp به صورت جداگانه‌ای در فایل pdf نهایی ثبت خواهد شد.

این مورد خصوصاً در تهیه گزارشاتی که تصویری را در پشت صفحه صفحات نمایش می‌دهد یا در هدر صفحه یک تصویر مشخص و ثابتی قرار گرفته است و نیاز است این تصویر در تمام صفحات تکرار شود، بسیار مهم است و در صورت عدم رعایت نکته تهیه یک

وهله از تصاویری تکراری، می تواند حجم فایل را بی جهت تا چندمگابایت افزایش دهد.

نظرات خوانندگان

نویسنده: محمدرضا
تاریخ: ۹:۴۸ ۱۳۹۱/۰۸/۲۸

سلام

آیا این امکان برای تصویر به کاربرده شده در htmlheader هم فراهم است؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۲ ۱۳۹۱/۰۸/۲۸

مطلب فوق در مورد iTextSharp بود.

در PdfReport متد header.CacheHeader وجود دارد که کل هدر را کش می‌کند (حالت پیش فرض است) و اشیاء آن را یکبار محاسبه و به صفحات اضافه خواهد کرد (در آخرین نگارش موجود در SVN). به این ترتیب فرقی نمی‌کند که هدر سفارشی است یا هر نوع پیش فرض دیگری. برای تمام آن‌ها کش توکار اعمال خواهد شد. اگر خواستید به ازای صفحات مختلف هدرهای مختلفی داشته باشید نیاز است header.CacheHeader را false کنید. در این حالت بهینه سازی صورت نخواهد گرفت.