ALLAN THRAEN | 17 years ago | PDF |

OUTPUT EPISERVER PAGES AS PDF

Got a question today from a partner about how he could make a "Download this page as PDF" functionality on a customers web site. I suppose there's quite a lot of ways to do it (just as many as there are HTML -> PDF libraries out there) but I remembered having used the free iText.NET library before and decided that this would be a nice little code-sample to put up here. Especially since it both demonstrates the use of PagePlugins, and PDF generation. I would expect that now that PDF is an ISO standard this will be a desired functionality lots of places.

My goal: To build the functionality on all EPiServer pages on a given website that adding the parameter "*output=pdf*" to their query-line will return a PDF document generated on-the-fly with a few of their key properties.

First, I downloaded the iTextSharp port of iText.NET. It's a single assembly you can reference in your project, so it's a nice, clean and easy approach.

Secondly, I used the (relatively) new cool PagePlugin feature in EPiServer CMS 5 that allows me to provide code that should be executed whenever a page is displayed. In my plugin I hooked onto the PreRender event of the page, created a new PDF document and put the PageName property in a Paragraph at the top. In order to also put the "MainBody" property on the page I had to parse it, so I used the simple Htmlparser that comes with iTextSharp - but I suppose I could just as well have used the HtmlParser built into EPiServer if I wanted an improved handling of how the html was rendered as a PDF.

See the code here:

```
using EPiServer.PlugIn;
using iTextSharp.text;
using iTextSharp.text.pdf;
using EPiServer;
using iTextSharp.text.html;
using System.Xml;
[PagePlugIn()]
public class PdfExtension
{
    public static void Initialize(int bitflags)
    {
        EPiServer.PageBase.PageSetup += new EPiServer.PageSetupEventHandler(PageBase_PageSetup);
    }

    static void PageBase_PageSetup(EPiServer.PageBase sender, EPiServer.PageSetupEventArgs e)
    {
        //Do all the magic on the PreRender event in order to have the correct environment set up.
        sender.PreRender += new EventHandler(sender_PreRender);
    }

    static void sender_PreRender(object sender, EventArgs e)
    {
        //Handle if the url is ....?output=pdf
        if ((!string.IsNullOrEmpty(HttpContext.Current.Request["output"])) &&
            (HttpContext.Current.Request["output"].ToLower() == "pdf"))
        {
            //Prepare to output a PDF
            HttpContext.Current.Response.Clear();
            HttpContext.Current.Response.ContentType = "application/pdf";

            //Build the PDF document
            Document d = new Document(PageSize.A4, 80, 50, 30, 65);
            PdfWriter pw = PdfWriter.GetInstance(d, HttpContext.Current.Response.OutputStream);
            d.Open();

            //Add the first line with the Page Name
            d.Add(new Paragraph((sender as PageBase).CurrentPage.PageName));

            //Load and parse the MainBody property. Enclosed in <body> tags to produce xhtml.
            XmlDocument xd = new XmlDocument();
            xd.LoadXml("<body>" + (sender as PageBase).CurrentPage["MainBody"].ToString() + "</body>")
            HtmlParser.Parse(d, xd);

            d.Close();
            //End output
            HttpContext.Current.Response.End();
        }
    }

}
```

CVR: 39680688