DOWNLOAD PDF C ITEXTSHARP EXAMPLES

Chapter 1 : calendrierdelascience.comeState() Example

iTextSharp is open source PDF solution. In most of the examples below, I tried to alter,copy a template PDF and then save it into a brand new output PDF file.

AddTitle "Hello World example" ; doc. AddKeywords "Metadata, iTextSharp 5. AddAuthor "Debopam Pal" ; doc. Creating a Multipage Document: We can create a new page through iTextSharp. Lets add five pages in PDF Document: Add new Paragraph string. Add new Paragraph "Hi! PdfStamper object to write Data from iTextSharp. BeginLayer layer ; cb. SetGState gState ; cb. So if I remove this Layer we can remove the content of the Layer also e. Now follow the Steps below to remove the Watermark Text from Layer: Read the existing watermarked document using iTextSharp. Taking the Content of the Page in the iTextSharp. If found then remove it by giving it zero length and zero data using two methods: Now, we already know that, watermark cannot be add during Page creation, it have to add after document creation. The events are following: Add new Paragraph "This is a page 1" ; doc. Add new Paragraph "Hello World" ; document. GetBuffer , 0, ms. The values of the different ViewerPreferences were originally stored in iTextSharp. PdfWriter class as an integer constant. You can set the ViewerPreferences by following two ways: By setting property ViewerPreferences of iTextSharp. To know all the ViewerPreferences and its purpose, please read this first. To know which value is appropiate for which key, read this first. By SetEncryption method of iTextSharp. Read full documentation of this method here. To know all the encryption types, click here. If any doubt, just post your comment below.

Chapter 2 : c# - Where download examples of itextsharp? - Stack Overflow

The original C# examples are hosted on Sourceforge. Unfortunately, Sourceforge suffered from a major outage a couple of weeks ago and they have not fully restored every service. This means that the examples can't be reached for the moment.

Height - heightOffset; document. GetInstance document, output ; writer. Height - heightOffset; iTextSharp. GetInstance images[i] ; pool. Close ; return output. GetInstance document, byteStream ; document. FromImage awtImg ; g. FillRectangle new SolidBrush Color. Green , 10, 10, 80, 80 ; g. GetInstance awtImg, BaseColor null ; document. Add itextImg ; document. GetInstance document, fileStream ; document. GetFont "Arial", 19, Font. Add mainTable ; document. AddKeywords "estimation" ; document. GetRequiredString "action" ; context. FindView context, ViewName, null. GetInstance document, memoryStream ; instance. TempData, writer ; view. ParseXHtml instance, document, inp, Stream null ; document. GetInstance document, outputStream ; document. GetPageSizeWithRotation i ; document. AddTemplate page, 0, -1f, 1f, 0, 0, reader. Close ; return outputStream. ToString "yyyyMMddhhmmss" , ". SetPageSize pagesize ; image. SetAbsolutePosition 0f, 0f ; doc. GetInstance document, stream ; document. SetMargins 36, 72, , ; document. Add new Paragraph "The left margin of this odd page is 36pt 0. Add paragraph ; document. Add new Paragraph "The right margin of this even page is 36pt 0. A3, 0, 0, 0, 0 ; document. GetInstance pdfDoc, outputStream ; pdfDoc. GetImportedPage reader, page ; pdfDoc. GetInstance doc, new System. AddTitle "My Science Report" ; doc. AddPageWithBasicFormatting doc ; this. AddPageWithInternalLinks doc ; this. AddPageWithBulletList doc ; this. AddPageWithExternalLinks doc ; this. SetMargins 14f, 0f, 3. FetchIndividualList sort, qid ; break; case "FamilyMembers": FetchFamilyList sort, qid ; break; case "ParentsOf": Address ; if m. FmtFone "C " , m. MimeType uri ; if mime! SetMargins 0, 0, 0, 0 ; document. SetPageSize new Rectangle logo. SetMargins 0, 0, 0, 0 ; PdfWriter. Rectangle 0, 0, sourceImage. SetPageSize pageSize ; document. GetInstance sourceFilePath ; document. SetMarginMirroringTopBottom true ; document. Add new Paragraph "Hello World" ; document. GetImportedPage reader, i ; cb. Close ; if outputCopy! Close ; if File. GetInstance document, output ; document.

Chapter 3 : Simple .Net Solutions | Adventures in Problem Solving

I am trying to use PdfSmartCopy from ItextSharp but I cannot find any relevant examples in c#.. The ideea is that I have a pdf containing form fields and the fields add kb to the size of the pdf document.

Download demo project - 1. The example project uses an ASP. NET MVC3 application to demonstrate the code but it should not be difficult to adapt it to your own needs. Background Me and some friends have been running a hobby web-site called malleus. One of the applications on our site is a "character generator" where users can build, configure, and maintain their role-playing characters. We wanted to expand this tool with a print functionality allowing our users to have their characters printed to a nice looking character sheet. We decided that we really needed to generate a PDF document. PDF documents are supported in all browsers and seems to be the de-facto standard for documents on the web. There are a lot of PDF components available on the net, but we quickly settled on iText as it is a very mature product. I encourage you to check out the details at http: The example creates a "diploma" for a bicycle race. You can enter name, date, the name of the race, and the distance. The web application will produce a diploma with the entered text on top. The basic flow of the example application is illustrated with images in figure 1, 2, and 3: Even though the diploma application is very basic, it contains all the programmatic elements required for producing output that looks like figure 4. The artistic skill that goes into images and layout is a job for a designer. In your Visual Studio project, you need to reference the itextsharp. Document generation strategy When you open up iText and begin examining the API, you will notice that it is capable of writing text, drawing figures, inserting images, creating PDF forms, etc.. For our task, we decided to use a combination of image fragments and text. The bottom layer will contain all the image fragments and the top layer will contain the text. It provides various document level information such as title, page count, etc.: GetInstance document, output ; document. Create memoryStream ; return File memoryStream. To actually fill the document with content, you need to learn to use the PdfWriter, PdfReader, and PdfTemplate classes. Bottom layer The bottom layer of the page is filled with other PDF fragments and images. Figure 2 contains our background image. To load these elements into the document, you do the following: Use the reader to create a PdfTemplate instance. Add the PdfTemplate to your current Document instance. AddTemplate background, 0, 0 ; Resource. GetBicycle simply returns a Stream instance to a resource embedded within the assembly. The call to document. NewPage generates a new page in the Document instance and the writer. At first it seems odd that you have to use the PdfWriter instance to transform the contents of the PdfReader into a PdfTemplate. As Bruno Lowagie - the creator of iText - explained to me, this is done to ensure that shared objects are reused. For instance, if you embed your own fonts in the document, the best option is to reuse this font whenever you write some text. For more specific details, I encourage you to get a hold of the book "iText in Action - 2nd edition". Top layer The top layer is where I put the text. As with everything else in iText, you have several options. You can add entire paragraph objects or print text at specific points. Flush ; The numbers in this code fragment represent a position in a coordinate system. Setting the font and size: This will affect the invocations to the following method: Closing the document To end the document, you should flush your writer and call Close on the document: Close ; This has the effect of pushing the PDF document down to the output stream that you used to create the document with in the first place. Tricks When generating documents this way, you often need to adjust the position of text and images. To help with this, I created two methods for adding a ruler to the generated PDF document. If you check the "add rulers" box in the example application, you should see something like the following: Figure 5 - Diploma with rulers Having these rulers makes it much easier to find the positions of text and images. You can also play around with the code-page being used by iText when generating text. This is done by changing the value of the encoding parameter of the BaseFont. The example uses the value "Cp" which is for Western Europe. Reading the source code for iText, I can see that it also supports the values "Cp" Central- and Eastern Europe and "Cp" the three Baltic states but I have yet to play around with these. Conclusion iText is a mature piece of software with tons of possibilities. I have only scratched the surface of what is possible. I once tried to read the PDF specification and found that it is one complex beast. I very much prefer delegating the work to an API such as iText. History 1 November - First edition.

Chapter 4 : iText PDF, easy PDF generation for Java or .NET developers | iText

This is the first of three articles about creating PDF documents using iTextSharp. The Namespace is really big, so I will focus on the parts you'll probably use when you need to create PDFs on a daily basis.

Create ; Open the document and add some content: Close does it all for you in one call: A simple PDF Of course, your requirements will rarely be so simple. In the previous example, we used a paragraph to add text to the document, but you can also use phrases and chunks to create the text you want. A chunk is simply any piece of text with a consistent style; using it, you can specify independent fonts and colors. A phrase is a collection of chunks that includes a leading separator the amount of vertical space between lines. Chunks and phrases can be added to paragraphs or added directly to documents. GetFont "Tahoma", 14 ; and then create a paragraph and add those chunks: Chunks with fonts iTextSharp also provides support for working with images and embedding those images in your documents. Images can be added through URLs: You can now add the image to the document or paragraph with the following code: Add image ; pdfDocument. Add p ; pdfDocument. Create and work with tables Create headers and footers Create anchors, lists, and annotations Manipulating PDFs with iTextSharp On a recent project, I was given the requirement of assembling large policy packets for an insurance client. The client had each piece of the policy either in a static PDF or in a file that would be dynamically converted to PDF. We wanted to get to an end result where the packet would be a single PDF, so it would be easier to store and would require only one print job, so there was no chance of part of it being lost. The problem was that the packet was different based on each individual policy, so we had to find a way to dynamically combine anywhere from 3 to 20 PDFs into one document. You can compile the code as a Console application and use it as-is, or you can modify it and use it directly in your application. With Safari, you learn the way you learn best. Get unlimited access to videos, live online training, learning paths, books, interactive tutorials, and more.

Chapter 5 : calendrierdelascience.comxtAligned, calendrierdelascience.com C# (CSharp) Method Code Ex

For example here is one of the newer articles I found out there: Create/Read Advance PDF Report using iTextSharp in C#.NET: Part I Also if you have an calendrierdelascience.com application I would recommend you to check out this solution for creating PDF files in C#.

Chapter 6 : iTextSharp: PdfPTable Basic Example â€“ Justin Cooney

C# (CSharp) Method calendrierdelascience.comeSize Code Examples This page contains top rated real world C# (CSharp) examples of method calendrierdelascience.comeSize extracted from open source projects.

Chapter 7 : C# tutorial: draw graphics in PDF document

C# (CSharp) Method calendrierdelascience.comxtAligned Code Examples This page contains top rated real world C# (CSharp) examples of method calendrierdelascience.comxtAligned extracted from open source projects.

Chapter 8 : How to generate PDF file using iTextSharp in C#

Yes, there are a few examples in C#, but that is not sufficient to the Beginners and the examples are in older version of iTextSharp, there are lots of change in the latest version. So, it will be difficult for beginners to convert older version to latest version.

Chapter 9 : calendrierdelascience.comeSize, calendrierdelascience.com C# (CSharp) Method Code Exam

DOWNLOAD PDF C ITEXTSHARP EXAMPLES

The examples contained herein are dependent upon the availability of the iTextSharp DLL; use the link provided previously in order to download the DLL locally to your development machine. In order to demonstrate filling out a PDF using the iTextSharp DLL, I downloaded a copy of the W-4 PDF form from the IRS website.