Search text in PDF documents using regular expressions

Written by Apitron Documentation Team

Introduction

Searching text in PDF document is easy and this feature became available to users of our Apitron PDF Rasterizer for .NET component many releases ago. Now we've updated the API and you can search for text on PDF page using standard .NET regular expression objects (Regex).

Text search API offered by Apitron PDF Rasterizer is decoupled from the rendering part and can be used independently. It's represented by the `SearchIndex` class that handles all search tasks and offers very useful features like building search indices for the documents, and saving/loading of such indices for the later use.

Using search API offered by Apitron PDF Rasterizer you can also highlight text on rendered pages because you get all necessary information about text position on PDF page.

See the code section for details.

The code

```
class Program
{
    // global rendering settings
    static RenderingSettings renderingSettings = new RenderingSettings();
    // hightlight brush for search results
    static Brush hightlightBrush = new SolidBrush(Color.FromArgb(100,255,255,0));

    static void Main(string[] args)
    {
        // the source file to search the text into
        string inputFilePath = "../../data/Apitron_Pdf_Kit_in_Action.pdf";

        // open pdf document for search and rendering
        // we'll use 2 different streams here
        using (Stream searchStream = new FileStream(inputFilePath, FileMode.Open,
            FileAccess.Read),
            documentStream = new FileStream(inputFilePath, FileMode.Open, FileAccess.Read))
        {
            // create search object from PDF data stream
            using (SearchIndex searchIndex = new SearchIndex(searchStream))
            {
                // open document to be used for rendering
                using (Document doc = new Document(documentStream))
                {
                    searchIndex.Search((handlerArgs =>
                    {
                        // if we have results
                        if (handlerArgs.ResultItems.Count != 0)
                        {
                            // create resulting image filename
                            string outputFileName = string.Format("{0}_{1}.png",
                                Path.GetFileNameWithoutExtension(inputFilePath),
                                handlerArgs.PageIndex);

                            // render found result and start system image viewer
                            Page page = doc.Pages[handlerArgs.PageIndex];
                            using (Image bitmap = page.Render(new Resolution(96, 96),
                                renderingSettings))
                            {
                                foreach (SearchResultItem searchResultItem in
                                    handlerArgs.ResultItems)
                                {
                                    HighlightSearchResult(bitmap, searchResultItem, page);
                                }

                                bitmap.Save(outputFileName);
                            }

                            Process.Start(outputFileName);
                        }

                    }),
                    // find everything that matches [WORD][whitespaces]Kit pattern
                    new Regex("\\w+\\s+Kit"));
                }
            }
        }
    }
```

```
    /// <summary>
    /// Highlights the search result.
    /// </summary>
    /// <param name="bitmap"> The bitmap. </param>
    /// <param name="searchResultItem"> The search result item. </param>
    /// <param name="page"> The page. </param>
    private static void HighlightSearchResult(Image bitmap, SearchResultItem searchResultItem,
        Page page)
    {
        using (Graphics gr = Graphics.FromImage(bitmap))
        {
            double[] rectangle;
            SearchResultRegion region = page.TransformRegion(searchResultItem.Region,
                bitmap.Width, bitmap.Height, renderingSettings);

            foreach (double[] item in region.Blocks)
            {
                rectangle = item;
                PointF[] points = new PointF[rectangle.Length / 2];
                for (int i = 0; i < 4; i++)
                {
                    points[i] = new PointF((float)rectangle[i * 2],
                        (float)rectangle[(i * 2) + 1]);
                }

                gr.FillPolygon(hightlightBrush, points);
            }
        }
    }
}
```

The complete code sample can be found in our [github] repo.

Results of the execution are shown below; please note that in evaluation mode search API looks for text on first three pages only.

1. Introduction

This book is intended to give you deep knowledge and understanding of the development techniques needed to properly use the Apitron PDF Kit for .NET component for PDF manipulation tasks. It includes overview of PDF generation, editing and signing as well as various content extraction operations you may need to complete during the development of your application.

The component itself has rich API and is suitable for cross-platform development being .NET, MONO and Xamarin compatible. It was widely adopted and is being used by many desktop, server and mobile applications on the market. Novice and experienced developers could both benefit from reading this book, because even if some concepts of PDF might be known and parts of the API might look familiar there are totally new things that needs to be mastered, e.g. Flow PDF generation API.

The book is based on version 1.0.23 of the Apitron PDF Kit for .NET and all samples provided here were created and tested using this version.

Pic. 1 Search text in PDF document - highlighted text

Summary

Apitron PDF Rasterizer for .NET is a complex solution that you can use for PDF rendering and also for implementing text search in PDF documents. It's a cross-platform library available for many .NET based platforms (Xamarin, Mono, .NET just to name a few) and can be used to create mobile, desktop and web applications. Contact us if you have any questions regarding our products or services.