How to find keyword in PDF and extract page in VB.NET using ByteScout PDF Extractor SDK

Write code in VB.NET to find keyword in PDF and extract page with this step-by-step tutorial

Every ByteScout tool contains example VB.NET source codes that you can find here or in the folder with installed ByteScout product. ByteScout PDF Extractor SDK can find keyword in PDF and extract page. It can be used from VB.NET. ByteScout PDF Extractor SDK is the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction.

You will save a lot of time on writing and testing code as you may just take the VB.NET code from ByteScout PDF Extractor SDK for find keyword in PDF and extract page below and use it in your application. In order to implement the functionality, you should copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. You can use these VB.NET sample examples in one or many applications.

ByteScout PDF Extractor SDK free trial version is available on our website. VB.NET and other programming languages are supported.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout PDF Extractor SDK

Explore API Documentation

Get Free Training for ByteScout PDF Extractor SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Program.vb

```
' This example page extraction by found keyword.

Imports Bytescout.PDFExtractor

Class Program

    Friend Shared Sub Main(args As String())

        Dim inputFile As String = ".\sample2.pdf"

        ' Create Bytescout.PDFExtractor.TextExtractor instance
        Dim extractor As New TextExtractor()
        extractor.RegistrationName = "demo"
        extractor.RegistrationKey = "demo"

        ' Load sample PDF document
        extractor.LoadDocumentFromFile(inputFile)

        Dim pageCount As Integer = extractor.GetPageCount()

        ' Search each page for a keyword
        For i As Integer = 0 To pageCount - 1

            If extractor.Find(i, "bombardment", False) Then

                ' Extract page
                Using splitter As New DocumentSplitter("demo", "demo")

                    splitter.OptimizeSplittedDocuments = True

                    Dim pageNumber As Integer = i + 1
                    ' (!) page number in ExtractPage() is 1-based
                    Dim outputFile As String = ".\page" & pageNumber.ToString() & ".pd
                    splitter.ExtractPage(inputFile, outputFile, pageNumber)

                    Console.WriteLine("Extracted page " & pageNumber.ToString() & " to

                End Using
            End If
        Next

        ' Cleanup
                extractor.Dispose()

        Console.WriteLine()
        Console.WriteLine("Press any key...")
        Console.ReadKey()

    End Sub
End Class
```

VIDEO

https://www.youtube.com/watch?v=s28W3_KMraU

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout PDF Extractor SDK Home Page
Explore ByteScout PDF Extractor SDK Documentation
Explore Samples
Sign Up for ByteScout PDF Extractor SDK Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co