www.bytescout.com

How to find text in PDF with smart match in VB.NET using ByteScout PDF Extractor SDK

The tutorial below will demonstrate how to find text in PDF with smart match in VB.NET

Source code documentation samples provide quick and easy way to add a required functionality into your application. What is ByteScout PDF Extractor SDK? It 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. It can help you to find text in PDF with smart match in your VB.NET application.

This rich sample source code in VB.NET for ByteScout PDF Extractor SDK includes the number of functions and options you should do calling the API to find text in PDF with smart match. In your VB.NET project or application you may simply copy & paste the code and then run your app! Test VB.NET sample code examples whether they respond your needs and requirements for the project.

Free trial version of ByteScout PDF Extractor SDK is available on our website. Documentation and source code samples are included.

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

```
Imports Bytescout.PDFExtractor

Class Program

    Friend Shared Sub Main(args As String())

        Dim extractor As TextExtractor = New TextExtractor("demo", "demo")

        ' Load the document
        extractor.LoadDocumentFromFile("sample2.pdf")

        ' Smart match the search string like Adobe Reader
        extractor.WordMatchingMode = WordMatchingMode.SmartMatch

        Dim searchString As String = "land"

        ' Get page count
        Dim pageCount As Integer = extractor.GetPageCount()

        ' Iterate through pages
        For i As Integer = 0 To pageCount - 1

            ' Search through page
            If extractor.Find(i, searchString, False) Then

                Do
                    ' Output search results
                    Console.WriteLine("Found on page " + i.ToString() + " at location "

                    ' Now we are getting the found text
                    Dim extractedString As String = extractor.FoundText.Text
                    Console.WriteLine("Found text: " + extractedString)

                Loop While extractor.FindNext() ' Search next occurrence of the search

            End If

        Next

        ' Cleanup
                extractor.Dispose()


        Console.WriteLine()
        Console.WriteLine("Press any key to exit...")
        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