How to PDF text search API in VB.NET with ByteScout Cloud API Server

Learning is essential in computer world and the tutorial below will demonstrate how to PDF text search API in VB.NET

The sample source code below will teach you how to PDF text search API in VB.NET. ByteScout Cloud API Server: the ready to use Web API Server that can be deployed in less than 30 minutes into your own in-house server or into private cloud server. Can store data on in-house local server based storage or in Amazon AWS S3 bucket. Processing data solely on the server using buil-in ByteScout powered engine, no cloud services are used to process your data!. It can PDF text search API in VB.NET.

These VB.NET code samples for VB.NET guide developers to speed up coding of the application when using ByteScout Cloud API Server. IF you want to implement the functionality, just copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. Use of ByteScout Cloud API Server in VB.NET is also described in the documentation included along with the product.

You can download free trial version of ByteScout Cloud API Server from our website with this and other source code samples for VB.NET.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Cloud API Server

Explore API Documentation

Get Free Training for ByteScout Cloud API Server

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

ByteScoutWebApiExample.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.26730.10
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "ByteScoutWebApiExample", "ByteScou
EndProject
Global
	GlobalSection(SolutionConfigurationPlatforms) = preSolution
		Debug|Any CPU = Debug|Any CPU
		Release|Any CPU = Release|Any CPU
	EndGlobalSection
	GlobalSection(ProjectConfigurationPlatforms) = postSolution
		{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.ActiveCfg = Debug
		{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.Build.0 = Debug|Ar
		{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.ActiveCfg = Rele
		{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.Build.0 = Releas
	EndGlobalSection
	GlobalSection(SolutionProperties) = preSolution
		HideSolutionNode = FALSE
	EndGlobalSection
	GlobalSection(ExtensibilityGlobals) = postSolution
		SolutionGuid = {4576C9BB-A42D-46A8-9198-7E2982E122FA}
	EndGlobalSection
EndGlobal
```

Module1.vb

```
Imports System.IO
Imports System.Net
Imports System.Threading
Imports Newtonsoft.Json.Linq

' Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://loc
' If it's not then please replace this with with your hosting url.
Module Module1

    ' Direct URL of source PDF file.
    Const SourceFileUrl As String = "https://bytescout-com.s3.amazonaws.com/files/demo-

    ' Comma-separated list of page indices (or ranges) to process. Leave empty for all
    Const Pages As String = ""

    ' PDF document password. Leave empty for unprotected documents.
```

```
Const Password As String = ""

' Search string.
Const SearchString As String = "\d{1,}\.\d\d" 'Regular expression To find numbers 
' Note: Do Not use `+` char in regex, but use `{1,}` instead.
' `+` char Is valid for URL And will Not be escaped, And it will become a space ch

' Enable regular expressions (Regex)
Const RegexSearch As Boolean = True

'(!) Make asynchronous job
Const Async As Boolean = True

Sub Main()

    ' Create standard .NET web client instance
    Dim webClient As WebClient = New WebClient()

    ' Prepare URL for PDF text search API call.
    Dim query As String = Uri.EscapeUriString(
        String.Format("https://localhost/pdf/find?password={0}&pages={1}&url={2}&s
            Password,
            Pages,
            SourceFileUrl,
            SearchString,
            RegexSearch,
            Async))

    Try
        ' Execute request
        Dim response As String = webClient.DownloadString(query)

        ' Parse JSON response
        Dim json As JObject = JObject.Parse(response)

        If json("error").ToObject(Of Boolean) = False Then

            ' Asynchronous job ID
            Dim jobId As String = json("jobId").ToString()

            ' URL of generated json file that will available after the job complet
            Dim resultFileUrl As String = json("url").ToString()

            ' Check the job status in a loop.
            ' If you don't want to pause the main thread you can rework the code
            ' to use a separate thread for the status checking And completion.
            Do
                Dim status = CheckJobStatus(jobId) ' Possible statuses: "working",

                ' Display timestamp and status (for demo purpose)
                Console.WriteLine(DateTime.Now.ToLongTimeString() + ": " + status)

                If (status = "success") Then
                    ' Execute request
                    Dim respFileJson As String = webClient.DownloadString(resultFi

                    ' Parse JSON response
                    Dim jsonFoundInformation As JArray = JArray.Parse(respFileJson

                    ' Display found information in console
```

```
                        For Each item As JToken In jsonFoundInformation
                            Console.WriteLine({code}quot;Found text {item("text")} at 
                        Next

                        Exit Do
                    ElseIf (status = "working") Then
                        ' Pause for a few seconds
                        Thread.Sleep(3000)
                    Else
                        Console.WriteLine(status)
                        Exit Do
                    End If
                Loop
            Else
                Console.WriteLine(json("message").ToString())
            End If

        Catch ex As WebException
            Console.WriteLine(ex.ToString())
        End Try

        webClient.Dispose()


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

    End Sub


    Function CheckJobStatus(ByVal jobId As String)

        Using webClient As New WebClient

            Dim url As String = "https://localhost/job/check?jobid=" & jobId

            Dim response As String = webClient.DownloadString(url)
            Dim json As JObject = JObject.Parse(response)

            Return Convert.ToString(json("status"))

        End Using

    End Function

End Module
```

packages.config

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<packages>
  <package id="Newtonsoft.Json" version="10.0.3" targetFramework="net40" />
</packages>
```

VIDEO

https://www.youtube.com/watch?v=NEwNs2b9YN8

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Cloud API Server Home Page
Explore ByteScout Cloud API Server Documentation
Explore Samples
Sign Up for ByteScout Cloud API Server 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