

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

Learn to PDF text search API in VB.NET

ByteScout simple and easy to understand tutorials are planned to describe the code for both VB.NET beginners and advanced programmers. ByteScout Cloud API Server is 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 built-in ByteScout powered engine, no cloud services are used to process your data!. It can be applied to PDF text search API using VB.NET.

The following code snippet for ByteScout Cloud API Server works best when you need to quickly PDF text search API in your VB.NET application. Just copy and paste the code into your VB.NET application's code and follow the instructions. Use of ByteScout Cloud API Server in VB.NET is also described in the documentation included along with the product.

Our website gives trial version of ByteScout Cloud API Server for free. It also includes documentation and source code samples.

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](http://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", "ByteScoutWebApiExample.csproj", "{...}"
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|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.Build.0 = Release|Any CPU
    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 Newtonsoft.Json.Linq

' Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://localhost:5000/"
' 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-files/cloud-api/pdf-rendering/sample.pdf"

    ' Comma-separated list of page indices (or ranges) to process. Leave empty for all pages.
    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 char

' Enable regular expressions (Regex)
Const RegexSearch 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}&search={3}&regex={4}",
            Password,
            Pages,
            SourceFileUrl,
            SearchString,
            RegexSearch))

    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

            For Each item As JToken In json("body")
                Console.WriteLine($"{code}quot;Found text {item("text")} at coordinate {item("x")}, {item("y")}")
            Next

        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

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](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

[www.bytescout.com](http://www.bytescout.com)