

How to PDF text search API in VB.NET with PDF.co Web API

The tutorial below will demonstrate how to PDF text search API in VB.NET

Every ByteScout tool contains example VB.NET source codes that you can find here or in the folder with installed ByteScout product. What is PDF.co Web API? It is the Web API with a set of tools for documents manipulation, data conversion, data extraction, splitting and merging of documents. Includes image recognition, built-in OCR, barcode generation and barcode decoders to decode bar codes from scans, pictures and pdf. It can help you to PDF text search API in your VB.NET application.

This rich sample source code in VB.NET for PDF.co Web API includes the number of functions and options you should do calling the API to PDF text search API. Just copy and paste the code into your VB.NET application's code and follow the instruction. Use of PDF.co Web API in VB.NET is also explained in the documentation included along with the product.

Trial version of PDF.co Web API can be downloaded for free from our website. It also includes source code samples for VB.NET and other programming languages.

VB.NET - Module1.vb

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

Module Module1

    ' The authentication key (API Key).
    ' Get your own by registering at https://app.pdf.co/documentation/api
    Const API_KEY As String = "*****"

    ' Source PDF file
    Const SourceFile As String = ".\sample.pdf"

    ' Comma-separated list of page indices (or ranges) to process. Leave empty for
    all pages. Example: '0,2-5,7-'.
    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
    Like '100.00'
    ' 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 on the server side.
```

```
' 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()
```

```
' Set API Key  
webClient.Headers.Add("x-api-key", API_KEY)
```

```
' 1. RETRIEVE THE PRESIGNED URL TO UPLOAD THE FILE.  
' * If you already have a direct file URL, skip to the step 3.
```

```
' Prepare URL for `Get Presigned URL` API call  
Dim query As String = Uri.EscapeUriString(String.Format(  
    "https://api.pdf.co/v1/file/upload/get-presigned-url?  
contenttype=application/octet-stream&name={0}",  
    Path.GetFileName(SourceFile)))
```

```
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
```

```
' Get URL to use for the file upload  
Dim uploadUrl As String = json("presignedUrl").ToString()  
' Get URL of uploaded file to use with later API calls  
Dim uploadedFileUrl As String = json("url").ToString()
```

```
' 2. UPLOAD THE FILE TO CLOUD.
```

```
webClient.Headers.Add("content-type", "application/octet-stream")  
webClient.UploadFile(uploadUrl, "PUT", SourceFile) ' You can use
```

```
UploadData() instead if your file is byte array or Stream
```

```
' 3. MAKE UPLOADED PDF FILE SEARCHABLE
```

```
' Prepare URL for PDF text search API call.
```

```
' See documentation: https :
```

```
//app.pdf.co/documentation/api/1.0/pdf/find.html
```

```
query = Uri.EscapeUriString(  
    String.Format("https://api.pdf.co/v1/pdf/find?password=  
{0}&pages={1}&url={2}&searchString={3}@exSearch={4}&async={5}",  
        Password,  
        Pages,  
        uploadedFileUrl,  
        SearchString,  
        RegexSearch,  
        Async))
```

```
' Execute request
```

```

response = webClient.DownloadString(query)

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

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

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

    ' URL of generated json file that will available after the job
    completion
    Dim resultFileUrl = 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", "failed", "aborted", "success".

        ' 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(resultFileUrl)

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

            ' Display found information in console
            For Each item As JToken In jsonFoundInformation
                Console.WriteLine($"Found text {item("text")} at
            coordinates {item("left")}, {item("top")}")
            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

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

        ' Set API Key
        webClient.Headers.Add("x-api-key", API_KEY)

        Dim url As String = "https://api.pdf.co/v1/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
```

VB.NET - packages.config

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about PDF.co Web API](#)

[Explore documentation](#)

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

or

[Get Your Free API Key for www.PDF.co Web API](#)