

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

Write code in VB.NET to PDF text search API with this step-by-step tutorial

These source code samples are listed and grouped by their programming language and functions they use. What is PDF.co Web API? It is the flexible Web API that includes full set of functions from e-signature requests to data extraction, OCR, images recognition, pdf splitting and pdf splitting. Can also generate barcodes and read barcodes from images, scans and pdf. It can help you to PDF text search API in your VB.NET application.

Fast application programming interfaces of PDF.co Web API for VB.NET plus the instruction and the code below will help you quickly learn how to PDF text search API. 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. Further enhancement of the code will make it more vigorous.

Free trial version of PDF.co Web API is available for download from our website. Get it to try other source code samples for VB.NET.

VB.NET - Module1.vb

```
Imports System.IO
Imports System.Net
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

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}",
                    Password,
                    Pages,
                    uploadedFileUrl,
                    SearchString,
                    RegexSearch))

            ' Execute request
            response = webClient.DownloadString(query)

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

```
        If json("error").ToObject(Of Boolean) = False Then
            For Each item As JToken In json("body")
                Console.WriteLine($"Found text {item("text")} at coordinates
{item("left")}, {item("top")}")
            Next
        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

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](https://www.pdf.co/web-api)