

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

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

Learn how to PDF text search API in VB.NET with this source code sample. PDF.co Web API 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 PDF text search API in VB.NET.

You will save a lot of time on writing and testing code as you may just take the VB.NET code from PDF.co Web API for PDF text search API below and use it in your application. 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 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 = "*****"

    ' Direct URL of source PDF file.
    Const SourceFileUrl As String = "https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-to-text/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)
```

```
' Prepare URL for PDF text search API call.  
' See documentation: https://app.pdf.co/documentation/api/1.0/pdf/find.html  
Dim query As String = 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,  
        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  
completion
```

```
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", "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 =
JSONArray.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

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

```



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](#)

or

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