

How to parse uploaded file for document parser API in VB.NET with PDF.co Web API

Learn in simple ways: How to parse uploaded file for document parser API in VB.NET

These source code samples are listed and grouped by their programming language and functions they use. Document parser API in VB.NET can be applied with PDF.co Web API. 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.

The SDK samples displayed below explain how to quickly make your application do document parser API in VB.NET with the help of PDF.co Web API. This VB.NET sample code can be used by copying and pasting into your project. Once done, just compile your project and click Run. Further improvement of the code will make it more robust.

PDF.co Web API - free trial version is available on our website. Also, there are other code samples to help you with your VB.NET application included into trial version.

FOR MORE INFORMATION AND FREE TRIAL:

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

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

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

[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.Collections.Specialized
Imports System.IO
Imports System.Net
Imports System.Text
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 = ".\MultiPageTable.pdf"

    ' PDF document password. Leave empty for unprotected documents.
    Const Password As String = ""

```

```

' Destination TXT file name
Const DestinationFile As String = ".\result.json"

Sub Main()

' Template text. Use Document Parser SDK (https://bytescout.com/products/development)
' to create templates.
' Read template from file
Dim templateText As String = File.ReadAllText("MultiPageTable-template1.yml")

' 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/pdf&filename={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() if you prefer

' 3. PARSE UPLOADED PDF DOCUMENT

' URL for `Document Parser` API call
query = "https://api.pdf.co/v1/pdf/documentparser"

Dim requestBody As New NameValueCollection()
requestBody.Add("url", uploadedFileUrl)
requestBody.Add("template", templateText)

' Execute request
Dim responseBytes As Byte() = webClient.UploadValues(query, "POST", requestBody)
response = Encoding.UTF8.GetString(responseBytes)

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

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

```

```

        ' Get URL of generated JSON file
        Dim resultFileUrl As String = json("url")

        ' Download JSON file
        webClient.DownloadFile(resultFileUrl, DestinationFile)

        Console.WriteLine("Generated JSON file saved as {0} file.", DestinationFile)
    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

```

MultiPageTable-template1.yml

```

---
# Template that demonstrates parsing of multi-page table using only
# regular expressions for the table start, end, and rows.
# If regular expression cannot be written for every table row (for example,
# if the table contains empty cells), try the second method demonstrated
# in `MultiPageTable-template2.yml` template.
templateVersion: 2
templatePriority: 0
sourceId: Multipage Table Test
detectionRules:
  keywords:
    - Sample document with multi-page table
  fields:
    total:
      expression: TOTAL {{DECIMAL}}
  tables:
    - name: table1
      start:
        # regular expression to find the table start in document
        expression: Item\s+Description\s+Price\s+Qty\s+Extended Price
      end:
        # regular expression to find the table end in document

```

```
expression: TOTAL\s+\d+\.\d\d
row:
  # regular expression to find table rows
  expression: '^s*(?<itemNo>\d+)\s+(?<description>.+)s+(?<price>\d+\.\d\d)s+(?<q
columns:
- name: itemNo
  type: integer
- name: description
  type: string
- name: price
  type: decimal
- name: qty
  type: integer
- name: extPrice
  type: decimal
multipage: true
```

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 PDF.co Web API Home Page](#)
[Explore PDF.co Web API Documentation](#)
[Explore Samples](#)
[Sign Up for PDF.co Web API 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