

# How to convert PDF to XLSX from uploaded file for PDF to excel API in VB.NET using PDF.co Web API

Tutorial: how to convert PDF to XLSX from uploaded file for PDF to excel API in VB.NET

ByteScout tutorials explain the code material for beginners and advanced programmers who are using VB.NET. PDF.co Web API helps with PDF to excel API in VB.NET. PDF.co Web API is the Rest API that provides set of data extraction functions, tools for documents manipulation, splitting and merging of pdf files. Includes built-in OCR, images recognition, can generate and read barcodes from images, scans and pdf.

Fast application programming interfaces of PDF.co Web API for VB.NET plus the instruction and the code below will help to learn how to convert PDF to XLSX from uploaded file. Sample code in VB.NET is all you need. Copy-paste it to your the code editor, then add a reference to PDF.co Web API and you are ready to try it! Test VB.NET sample code examples whether they respond your needs and requirements for the project.

Our website provides free trial version of PDF.co Web API that includes source code samples to help with your VB.NET project.

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 = ""
    ' Destination XLSX file name
    const DestinationFile as String = ".\result.xlsx"

    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. CONVERT UPLOADED PDF FILE TO XLSX

        ' Prepare URL for `PDF To XLSX` API call
        query = Uri.EscapeUriString(String.Format(
            "https://api.pdf.co/v1/pdf/convert/to/xlsx?
name={0}&password={1}&pages={2}&url={3}",
            Path.GetFileName(DestinationFile),
            Password,
            Pages,
            uploadedFileUrl))

        ' Execute request
        response = webClient.DownloadString(query)

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

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

            ' Get URL of generated XLSX file
            Dim resultFileUrl As string =
json("url").ToString()

```

```
        DestinationFile)
        ' Download XLSX file
        webClient.DownloadFile(resultFileUrl,
        DestinationFile)
        Console.WriteLine("Generated XLSX 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
```

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](#)