

How to convert PDF to XLS from URL asynchronously for PDF to excel API in VB.NET and PDF.co Web API

Step By Step Tutorial: how to convert PDF to XLS from URL asynchronously for PDF to excel API in VB.NET

The documentation is designed to help you to implement the features on your side. 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.

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 implement PDF to excel API. For implimentation of this functionality, please copy and paste code below into your app using code editor. Then compile and run your app. You can use these VB.NET sample examples in one or many applications.

PDF.co Web API - free trial version is on available 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", "{F184B08F-C81C-45F6-A57F-5ABD9991F28F}"
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.IO
Imports System.Net
Imports System.Threading
Imports Newtonsoft.Json.Linq

' Cloud API asynchronous "PDF To XLS" job example.
' Allows to avoid timeout errors when processing huge or scanned PDF documents.

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/example.pdf"
    ' Comma-separated list of page indices (or ranges) to process. Leave empty for all pages.
```

```

Const Pages As String = ""
' PDF document password. Leave empty for unprotected documents.
Const Password As String = ""
' Destination XLS file name
Const DestinationFile As String = ".\result.xls"
' (!) 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 To XLS` API call
    Dim query As String = Uri.EscapeUriString(String.Format(
        "https://api.pdf.co/v1/pdf/convert/to/xls?name={0}&password={1}&pages={2}&sourceFileUrl={3}&async={4}",
        Path.GetFileName(DestinationFile),
        Password,
        Pages,
        SourceFileUrl,
        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 XLS file that will be available after
            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 re
            ' to use a separate thread for the status checking and
            Do
                Dim status As String = CheckJobStatus(jobId)

                ' Display timestamp and status (for demo purposes)
                Console.WriteLine(DateTime.Now.ToLongTimeString() & " Status: " & status)

                If status = "success" Then

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

                    Console.WriteLine("Generated XLS file saved to " & DestinationFile)
                    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(jobId As String) As String

                    Using webClient As WebClient = 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=" +

                        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

```

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