

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

Tutorial: how to convert PDF to XML from URL asynchronously for PDF to XML API in VB.NET

Today we will explain the steps and algorithm of how to convert PDF to XML from URL asynchronously and how to make it work in your application. PDF.co Web API helps with PDF to XML API in VB.NET. PDF.co Web API 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.

The SDK samples like this one below explain how to quickly make your application do PDF to XML API in VB.NET with the help of PDF.co Web API. This VB.NET sample code should be copied and pasted into your project. After doing this just compile your project and click Run. Further enhancement of the code will make it more vigorous.

ByteScout free trial version is available for FREE download from our website. Programming tutorials along with source code samples are included.

VB.NET - Module1.vb

```
Imports System.IO
Imports System.Net
Imports System.Threading
Imports Newtonsoft.Json.Linq

' Cloud API asynchronous "PDF To XML" 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/demo-files/cloud-api/pdf-to-xml/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 XML file name
Const DestinationFile As String = ".\result.xml"
' (!) 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 XML` API call
    Dim query As String = Uri.EscapeUriString(String.Format(
        "https://api.pdf.co/v1/pdf/convert/to/xml?name={0}&password=
{1}&pages={2}&url={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 XML file that will 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
            ' to use a separate thread for the status checking
            ' and completion.
            Do
                Dim status As String = CheckJobStatus(jobId)
                ' Possible statuses: "working", "failed", "aborted", "success".
                ' Display timestamp and status (for demo
                ' purposes)
                Console.WriteLine(DateTime.Now.ToLongTimeString() + ": " + status)

                If status = "success" Then

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

                    Console.WriteLine("Generated XML file

```

```

saved as "{0}" file.", 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="
+ 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](#)