

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

How to convert PDF to HTML from URL asynchronously in VB.NET with easy ByteScout code samples to make PDF to HTML API. Step-by-step tutorial

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

VB.NET code snippet like this for PDF.co Web API works best when you need to quickly implement PDF to HTML API in your VB.NET application. Open your VB.NET project and simply copy & paste the code and then run your app! This basic programming language sample code for VB.NET will do the whole work for you in implementing PDF to HTML API in your app.

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 System.Threading
Imports Newtonsoft.Json.Linq

' Cloud API asynchronous "PDF To HTML" 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-html/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 HTML file name
Const DestinationFile As String = ".\result.html"
' Set to `true` to get simplified HTML without CSS. Default is the rich HTML
keeping the document design.
Const PlainHtml As Boolean = False
' Set to `true` if your document has the column layout like a newspaper.
Const ColumnLayout As Boolean = False
' (!) 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 HTML` API call
    Dim query As String = Uri.EscapeUriString(String.Format(
        "https://api.pdf.co/v1/pdf/convert/to/html?name={0}&password=
{1}&pages={2}&simple={3}&columns={4}&url={5}&async={6}",
        Path.GetFileName(DestinationFile),
        Password,
        Pages,
        PlainHtml,
        ColumnLayout,
        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 HTML file that will available
            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 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 HTML file
            WebClient.DownloadFile(resultFileUrl,
DestinationFile)

            Console.WriteLine("Generated HTML
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

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 \[www.PDF.co\]\(http://www.PDF.co\) Web API](#)