

How to convert web page to PDF from URL asynchronously for HTML to PDF API in VB.NET and PDF.co Web API

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

Writing of the code to convert web page to PDF from URL asynchronously in VB.NET can be done by developers of any level using PDF.co Web API. HTML to PDF API in VB.NET can be implemented with PDF.co Web API. 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.

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 HTML to PDF API. This VB.NET sample code should be copied and pasted into your project. After doing this just compile your project and click Run. Enjoy writing a code with ready-to-use sample VB.NET codes to add HTML to PDF API functions using PDF.co Web API in VB.NET.

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.

VB.NET - Module1.vb

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

' Cloud API asynchronous "Web Page to PDF" 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 = "*****"

    ' URL of web page to convert to PDF document.
    Const SourceUrl As String = "http://www.usa.gov"
    ' Destination PDF file name
    Const DestinationFile As String = ".\result.pdf"
    ' (!) 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 `Web Page to PDF` API call
    Dim query As String = Uri.EscapeUriString(String.Format(
        "{1}&async={2}",
        Path.GetFileName(DestinationFile),
        SourceUrl,
        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 PDF 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 PDF file
                    webClient.DownloadFile(resultFileUrl,
                    DestinationFile)

                    Console.WriteLine("Generated PDF 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 \[www.ByteScout.com\]\(http://www.ByteScout.com\)](#)

or

[Get Your Free API Key for \[www.PDF.co\]\(http://www.PDF.co\) Web API](#)