

How to split PDF from URL asynchronously for PDF splitting API in VB.NET with PDF.co Web API

Tutorial: how to split PDF from URL asynchronously for PDF splitting API in VB.NET

On this page you will learn from code samples for programming in VB.NET. PDF.co Web API was made to help with PDF splitting 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.

VB.NET code snippet like this for PDF.co Web API works best when you need to quickly implement PDF splitting API in your VB.NET application. For implementation of this functionality, please copy and paste code below into your app using code editor. Then compile and run your app. Enjoy writing a code with ready-to-use sample VB.NET codes to implement PDF splitting API using PDF.co Web API.

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 "Split 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 = "*****"

    ' Source PDF file to split
    Const SourceFileUrl As String = "https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-split/sample.pdf"
    ' Comma-separated list of page numbers (or ranges) to process. Example: '1,3-
5,7-'.
    Const Pages As String = "1-2,3-"
    ' (!) 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 `Split PDF` API call
    Dim query As String = Uri.EscapeUriString(String.Format(
        "https://api.pdf.co/v1/pdf/split?pages={0}&url={1}&async={2}",
        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 JSON file available after the job
            ' completion; it will contain URLs of result PDF files.
            Dim resultJsonFileUrl 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 JSON file as string
                    Dim jsonFileString As String =
                        webClient.DownloadString(resultJsonFileUrl)

                    Dim resultFileUrls As JArray =
                        JArray.Parse(jsonFileString)

                    ' Download generated PDF files
                    Dim part As Integer = 1
                    For Each token As JToken In
                        resultFileUrls

```

```

token.ToString()
String.Format(".\part{0}.pdf", part)

webClient.DownloadFile(resultFileUrl, localFileName)

""{0}"".", localFileName)

Dim resultFileUrl As String =
Dim localFileName As String =

Console.WriteLine("Downloaded
part = part + 1

Next
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](#)