

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

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

Source code documentation samples provide quick and easy way to add a required functionality into your application. PDF to image API in VB.NET can be implemented with PDF.co Web API. 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 samples for VB.NET developers help to speed up the application's code writing when using PDF.co Web API. Follow the instruction and copy - paste code for VB.NET into your project's code editor. Test VB.NET sample code examples whether they respond your needs and requirements for the project.

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 "PDF To PNG" 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
    Const SourceFileUrl As String = "https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-to-image/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 = ""
    ' (!) 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 PNG` API call  
    Dim query As String = Uri.EscapeUriString(String.Format(  
        "https://api.pdf.co/v1/pdf/convert/to/png?password={0}&pages=  
{1}&url={2}&async={3}",  
        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 JSON file available after the job  
            completion; it will contain URLs of result PNG 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 PNG files
```

```

Dim page As Integer = 1
For Each token As JToken In
resultFilesUrls
    Dim resultFileUrl As String =
token.ToString()
    Dim localFileName As String =
String.Format(".\page{0}.png", page)
webClient.DownloadFile(resultFileUrl, localFileName)
""{0}"".", localFileName)
    Console.WriteLine("Downloaded
page = page + 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](#)

or

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