

www.bytescout.com

How to generate PDF from large HTML file for HTML to PDF API in VB.NET and PDF.co Web API

PDF.co Web API is the Rest API that provides set of data extraction functions, tools for documents manipulation, splitting and merging of pdf files. Includes built-in OCR, images recognition, can generate and read barcodes from images, scans and pdf.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about PDF.co Web API](#)

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

Source Code Files:

ByteScoutWebApiExample.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.26730.10
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "ByteScoutWebApiExample", "ByteScoutWebApiExample
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.ActiveCfg = Release|Any CPU
```

```

    {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.Build.0 = Release|Any CPU
EndGlobalSection
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {4576C9BB-A42D-46A8-9198-7E2982E122FA}
EndGlobalSection
EndGlobal

```

ByteScoutWebApiExample.vbproj

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''">AnyCPU</Platform>
    <ProjectGuid>{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>ByteScoutWebApiExample.Module1</StartupObject>
    <RootNamespace>ByteScoutWebApiExample</RootNamespace>
    <AssemblyName>ByteScoutWebApiExample</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>>true</DefineDebug>
    <DefineTrace>>true</DefineTrace>
    <OutputPath>bin\Debug</OutputPath>
    <DocumentationFile>ByteScoutWebApiExample.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <DefineDebug>>false</DefineDebug>
    <DefineTrace>>true</DefineTrace>
    <Optimize>>true</Optimize>
    <OutputPath>bin\Release</OutputPath>
    <DocumentationFile>ByteScoutWebApiExample.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup>
    <OptionExplicit>On</OptionExplicit>
  </PropertyGroup>
  <PropertyGroup>
    <OptionCompare>Binary</OptionCompare>
  </PropertyGroup>
  <PropertyGroup>
    <OptionStrict>Off</OptionStrict>
  </PropertyGroup>
  <PropertyGroup>
    <OptionInfer>On</OptionInfer>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="Newtonsoft.Json, Version=10.0.0.0, Culture=neutral, PublicKeyToken=30ad4fe6b2a6aeed, p
      <HintPath>packages\Newtonsoft.Json.10.0.3\lib\net40\Newtonsoft.Json.dll</HintPath>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
  </ItemGroup>

```

```

<Reference Include="System.Xml" />
<Reference Include="System.Core" />
<Reference Include="System.Xml.Linq" />
<Reference Include="System.Data.DataSetExtensions" />
</ItemGroup>
<ItemGroup>
<Import Include="Microsoft.VisualBasic" />
<Import Include="System" />
<Import Include="System.Collections" />
<Import Include="System.Collections.Generic" />
<Import Include="System.Data" />
<Import Include="System.Diagnostics" />
<Import Include="System.Linq" />
<Import Include="System.Xml.Linq" />
</ItemGroup>
<ItemGroup>
<Compile Include="Module1.vb" />
</ItemGroup>
<ItemGroup>
<Content Include="sample.html">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</Content>
<None Include="packages.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
</Project>

```

Module1.vb

```

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

Module Module1

    ' The authentication key (API Key).
    ' Get your own by registering at https://app.pdf.co/documentation/api
    Const API_KEY As String = "*****!"

    ' HTML input
    Dim inputSample As String = File.ReadAllText(".\sample.html")

    ' Destination PDF file name
    Const DestinationFile As String = ".\result.pdf"

    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)

        ' Set JSON content type
        webClient.Headers.Add("Content-Type", "application/json")

        Try

            ' Prepare requests params as JSON
            ' See documentation: https://apidocs.pdf.co/?#1-json-pdfconvertfromhtml
            Dim parameters As New Dictionary(Of String, Object)

            ' Input HTML code to be converted. Required.
            parameters.Add("html", inputSample)

```

```

' Name of resulting file
parameters.Add("name", Path.GetFileName(DestinationFile))

' Set to css style margins Like 10 px Or 5px 5px 5px 5px.
parameters.Add("margins", "5px 5px 5px 5px")

' Can be Letter, A4, A5, A6 Or custom size Like 200x200
parameters.Add("paperSize", "Letter")

' Set to Portrait Or Landscape. Portrait by default.
parameters.Add("orientation", "Portrait")

' true by default. Set to false to disbale printing of background.
parameters.Add("printBackground", True)

' If large input document, process in async mode by passing true
parameters.Add("async", True)

' Set to HTML for header to be applied on every page at the header.
parameters.Add("header", "")

' Set to HTML for footer to be applied on every page at the bottom.
parameters.Add("footer", "")

' Convert dictionary of params to JSON
Dim jsonPayload As String = JsonConvert.SerializeObject(parameters)

' Prepare URL for `HTML to PDF` API call
Dim url As String = "https://api.pdf.co/v1/pdf/convert/from/html"

' Execute POST request with JSON payload
Dim response As String = webClient.UploadString(url, jsonPayload)

' 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 the job completion
    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 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
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

```
''' <summary>
```

```
''' Check Async Task Job Status
```

```
''' </summary>
```

```
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
```

packages.config

```
<?xml version="1.0" encoding="utf-8"?>  
<packages>  
  <package id="Newtonsoft.Json" version="10.0.3" targetFramework="net40" />  
</packages>
```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit PDF.co Web API Home Page](#)

[Explore PDF.co Web API Documentation](#)
[Explore Samples](#)
[Sign Up for PDF.co Web API Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)
[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit www.Bytescout.com](http://www.Bytescout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com