www.bytescout.com

How to merge any documents from urls asynchronously for PDF merging API in VB.NET with PDF.co Web API

PDF.co Web API: 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

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", "ByteSco
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
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.Build.0 = Debug|A
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.ActiveCfg = Rel
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.Build.0 = Relea
        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/200
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.pr
  <PropertyGroup>
    <Configuration Condition=" '$(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, PublicKeyT
      <HintPath>packages\Newtonsoft.Json.10.0.3\lib\net40\Newtonsoft.Json.dll</HintPat
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
    <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" />
    <Compile Include="My Project\AssemblyInfo.vb" />
    <Compile Include="My Project\Application.Designer.vb">
      <AutoGen>True</AutoGen>
      <DependentUpon>Application.myapp</DependentUpon>
    </Compile>
    <Compile Include="My Project\Resources.Designer.vb">
      <AutoGen>True</AutoGen>
      <DesignTime>True</DesignTime>
      <DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <Compile Include="My Project\Settings.Designer.vb">
```

```
      <AutoGen>True</AutoGen>
      <DependentUpon>Settings.settings</DependentUpon>
      <DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
  </ItemGroup>
  <ItemGroup>
    <EmbeddedResource Include="My Project\Resources.resx">
      <Generator>VbMyResourcesResXFileCodeGenerator</Generator>
      <LastGenOutput>Resources.Designer.vb</LastGenOutput>
      <CustomToolNamespace>My.Resources</CustomToolNamespace>
      <SubType>Designer</SubType>
    </EmbeddedResource>
  </ItemGroup>
  <ItemGroup>
    <None Include="My Project\Application.myapp">
      <Generator>MyApplicationCodeGenerator</Generator>
      <LastGenOutput>Application.Designer.vb</LastGenOutput>
    </None>
    <None Include="My Project\Settings.settings">
      <Generator>SettingsSingleFileGenerator</Generator>
      <CustomToolNamespace>My</CustomToolNamespace>
      <LastGenOutput>Settings.Designer.vb</LastGenOutput>
    </None>
    <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

' Cloud API asynchronous "Merge 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 = "**********************************"

        ' Direct URLs of files to merge. Supports documents, spreadsheets, images as s
        Dim SourceFiles As String() = {
                        "https://bytescout-com.s3-us-west-2.amazonaws.com/files/demo-f
                        "https://bytescout-com.s3-us-west-2.amazonaws.com/files/demo-f
                        "https://bytescout-com.s3-us-west-2.amazonaws.com/files/demo-f
                        "https://bytescout-com.s3-us-west-2.amazonaws.com/files/demo-f
                        "https://bytescout-com.s3-us-west-2.amazonaws.com/files/demo-f
```

```
        }
' 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)

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

        ' Prepare URL for `Merge Document` API call
        Dim url As String = "https://api.pdf.co/v1/pdf/merge2"

' Prepare requests params as JSON
' See documentation: https : //apidocs.pdf.co
Dim parameters As New Dictionary(Of String, Object)
        parameters.Add("name", Path.GetFileName(DestinationFile))
        parameters.Add("url", String.Join(",", SourceFiles))
        parameters.Add("async", Async)

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

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

                ' 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 r
                        ' to use a separate thread for the status checking and
                        Do
                                Dim status As String = CheckJobStatus(jobId) '

                                ' Display timestamp and status (for demo purpo
                                Console.WriteLine(DateTime.Now.ToLongTimeStrin

                                If status = "success" Then

                                        ' Download PDF file
                                        webClient.DownloadFile(resultFileUrl,

                                        Console.WriteLine("Generated PDF file
```

```
                                        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=" +

                        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

visit www.PDF.co