www.bytescout.com

How to async file upload and async merge PDF for PDF merging API in C# using ByteScout Cloud API Server

ByteScout Cloud API Server is the ready to deploy Web API Server that can be deployed in less than thirty minutes into your own in-house Windows server (no Internet connnection is required to process data!) or into private cloud server. Can store data on in-house local server based storage or in Amazon AWS S3 bucket. Processing data solely on the server using built-in ByteScout powered engine, no cloud services are used to process your data!.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about **ByteScout Cloud API Server**

Explore API Documentation

Get Free Training for ByteScout Cloud API Server

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

ByteScoutWebApiExample.csproj

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="12.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.co
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.pr
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{1E1C2C34-017E-4605-AE2B-55EA3313BE51}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>ByteScoutWebApiExample</RootNamespace>
    <AssemblyName>ByteScoutWebApiExample</AssemblyName>
    <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
    <FileAlignment>512</FileAlignment>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <Optimize>false</Optimize>
    <OutputPath>bin\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <ErrorReport>prompt</ErrorReport>
    <WarningLevel>4</WarningLevel>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="Newtonsoft.Json, Version=10.0.0.0, Culture=neutral, PublicKeyT
      <HintPath>packages\Newtonsoft.Json.10.0.3\lib\net40\Newtonsoft.Json.dll</HintPat
      <Private>True</Private>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data" />
    <Reference Include="System.Xml" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="Program.cs" />
    <Compile Include="Properties\AssemblyInfo.cs" />
  </ItemGroup>
  <ItemGroup>
    <None Include="packages.config" />
    <Content Include="sample1.pdf">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
    <Content Include="sample2.pdf">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
  </ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
```

```
  <!-- To modify your build process, add your task inside one of the targets below and
       Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>
```

ByteScoutWebApiExample.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 2013
VisualStudioVersion = 12.0.40629.0
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ByteScoutWebApiExample", "ByteSco
EndProject
Global
        GlobalSection(SolutionConfigurationPlatforms) = preSolution
                Debug|Any CPU = Debug|Any CPU
                Release|Any CPU = Release|Any CPU
        EndGlobalSection
        GlobalSection(ProjectConfigurationPlatforms) = postSolution
                {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Debug|Any CPU.ActiveCfg = Debug
                {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Debug|Any CPU.Build.0 = Debug|A
                {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Release|Any CPU.ActiveCfg = Rel
                {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Release|Any CPU.Build.0 = Relea
        EndGlobalSection
        GlobalSection(SolutionProperties) = preSolution
                HideSolutionNode = FALSE
        EndGlobalSection
EndGlobal
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Net;
```

```
using System.Threading;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace ByteScoutWebApiExample
{
        // Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "h
        // If it's not then please replace this with with your hosting url.
        class Program
        {
                // The authentication key (API Key).
                // Get your own by registering at https://app.pdf.co/documentation/api
                const String API_KEY = "**********************************";

                // Source PDF files
                static string[] SourceFiles = new string[] { @".\sample1.pdf", @".\sam
                // Destination PDF file name
                const string DestinationFile = @".\result.pdf";
        // (!) Make asynchronous job
        const bool Async = true;

        static void Main(string[] args)
                {
                        // Create standard .NET web client instance
                        WebClient webClient = new WebClient();

                        // Set API Key
                        webClient.Headers.Add("x-api-key", API_KEY);

                        // 1. UPLOAD FILES TO CLOUD

                        List<string> uploadedFiles = new List<string>();

                        try
                        {
                                foreach (string pdfFile in SourceFiles)
                                {
                                        // 1a. RETRIEVE THE PRESIGNED URL TO UPLOAD TH

                                        // Prepare URL for `Get Presigned URL` API cal
                                        string query = Uri.EscapeUriString(string.Form
                                                "https://localhost/file/upload/get-pre
                                                Path.GetFileName(pdfFile)));

                                        // Execute request
                                        string response = webClient.DownloadString(que

                                        // Parse JSON response
                                        JObject json = JObject.Parse(response);

                                        if (json["error"].ToObject<bool>() == false)
                                        {
                                                // Get URL to use for the file upload
                                                string uploadUrl = json["presignedUrl"
                                                // Get URL of uploaded file to use wit
                                                string uploadedFileUrl = json["url"].T

                                                // 1b. UPLOAD THE FILE TO CLOUD.

                                                webClient.Headers.Add("content-type",
```

```
                                        webClient.UploadFile(uploadUrl, "PUT",

                                        uploadedFiles.Add(uploadedFileUrl);
                                }
                                else
                                {
                                        Console.WriteLine(json["message"].ToSt
                                }
                        }

                        if (uploadedFiles.Count > 0)
                        {
                // 2. MERGE UPLOADED PDF DOCUMENTS

                // URL for `Merge PDF` API call
                string url = "https://localhost/pdf/merge";

                // Prepare requests params as JSON
                Dictionary<string, object> parameters = new Dictionary<string, obj
                parameters.Add("name", Path.GetFileName(DestinationFile));
                parameters.Add("url", string.Join(",", uploadedFiles));
                parameters.Add("async", Async);

                // Convert dictionary of params to JSON
                string jsonPayload = JsonConvert.SerializeObject(parameters);

                try
                {
                    // Execute POST request with JSON payload
                    string response = webClient.UploadString(url, jsonPayload);

                    // Parse JSON response
                    JObject json = JObject.Parse(response);

                    if (json["error"].ToObject<bool>() == false)
                    {
                        // Asynchronous job ID
                        string jobId = json["jobId"].ToString();
                        // URL of generated PDF file that will available after the
                        string resultFileUrl = json["url"].ToString();

                        // Check the job status in a loop.
                        // If you don't want to pause the main thread you can rewo
                        // to use a separate thread for the status checking and co
                        do
                        {
                            string status = CheckJobStatus(jobId); // Possible sta

                            // Display timestamp and status (for demo purposes)
                            Console.WriteLine(DateTime.Now.ToLongTimeString() + ":

                            if (status == "success")
                            {
                                // Download PDF file
                                webClient.DownloadFile(resultFileUrl, DestinationF

                                Console.WriteLine("Generated PDF file saved as \"{
                                break;
                            }
                            else if (status == "working")
```

```
                                {
                                    // Pause for a few seconds
                                    Thread.Sleep(3000);
                                }
                                else
                                {
                                    Console.WriteLine(status);
                                    break;
                                }
                            }
                            while (true);
                        }
                        else
                        {
                            Console.WriteLine(json["message"].ToString());
                        }
                    }
                    catch (WebException e)
                    {
                        Console.WriteLine(e.ToString());
                    }
                }
                        }
                        catch (WebException e)
                        {
                                Console.WriteLine(e.ToString());
                        }

                        webClient.Dispose();


                        Console.WriteLine();
                        Console.WriteLine("Press any key...");
                        Console.ReadKey();
                }


        /// <summary>
        /// Check job status
        /// </summary>
        static string CheckJobStatus(string jobId)
        {
            using (WebClient webClient = new WebClient())
            {
                // Set API Key
                webClient.Headers.Add("x-api-key", API_KEY);

                string url = "https://localhost/job/check?jobid=" + jobId;

                string response = webClient.DownloadString(url);
                JObject json = JObject.Parse(response);

                return Convert.ToString(json["status"]);
            }
        }
    }
}
```

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 ByteScout Cloud API Server Home Page
Explore ByteScout Cloud API Server Documentation
Explore Samples
Sign Up for ByteScout Cloud API Server 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

www.bytescout.com