How to merge PDF documents from urls asynchronously for PDF merging API in C# using PDF.co Web API

Learn how to merge PDF documents from urls asynchronously to have PDF merging API in C#

Today you are going to learn how to merge PDF documents from urls asynchronously in C#. PDF.co Web API was made to help with PDF merging API in C#. 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.

C# code snippet like this for PDF.co Web API works best when you need to quickly implement PDF merging API in your C# application. Open your C# project and simply copy & paste the code and then run your app! Writing C# application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Trial version of ByteScout is available for free download from our website. This and other source code samples for C# and other programming languages are available.

C# - Program.cs

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


// Cloud API asynchronous "Merge PDF" job example.
// Allows to avoid timeout errors when processing huge or scanned PDF documents.

namespace ByteScoutWebApiExample
{
        class Program
        {
                // The authentication key (API Key).
                // Get your own by registering at
https://app.pdf.co/documentation/api
                const String API_KEY = "**********************************";

                // Direct URLs of PDF files to merge
                static string[] SourceFiles = {
                        "https://bytescout-com.s3.amazonaws.com/files/demo-
files/cloud-api/pdf-merge/sample1.pdf",
```

```
                        "https://bytescout-com.s3.amazonaws.com/files/demo-
files/cloud-api/pdf-merge/sample2.pdf" };
                // 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);

                        // Prepare URL for `Merge PDF` API call
                        string query = Uri.EscapeUriString(string.Format(
                                "https://api.pdf.co/v1/pdf/merge?name={0}&url=
{1}&async={2}",
                                Path.GetFileName(DestinationFile),
                                string.Join(",", SourceFiles),
                                Async));

                        try
                        {
                                // Execute request
                                string response = webClient.DownloadString(query);

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

                                if (json["error"].ToObject() == false)
                                {
                                        // Asynchronous job ID
                                        string jobId = json["jobId"].ToString();
                                        // URL of generated PDF file that will
available after the job completion
                                        string resultFileUrl =
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
                                        {
                                                string status =
CheckJobStatus(jobId); // Possible statuses: "working", "failed", "aborted",
"success".

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

                                                if (status == "success")
                                                {
                                                        // Download PDF file
```

```
webClient.DownloadFile(resultFileUrl, DestinationFile);

                                                Console.WriteLine("Generated
PDF file saved as \"{0}\" file.", DestinationFile);
                                                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());
                }

                webClient.Dispose();


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

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

                        string url = "https://api.pdf.co/v1/job/check?jobid="
+ jobId;


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

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