

How to split PDF from URL asynchronously for PDF splitting API in C# and PDF.co Web API

How to split PDF from URL asynchronously for PDF splitting API in C#: How To Tutorial

We've created and updating regularly our sample code library so you may quickly learn PDF splitting API and the step-by-step process in C#. PDF.co Web API was made to help with PDF splitting API in C#. PDF.co Web API is the flexible Web API that includes full set of functions from e-signature requests to data extraction, OCR, images recognition, pdf splitting and pdf splitting. Can also generate barcodes and read barcodes from images, scans and pdf.

The SDK samples like this one below explain how to quickly make your application do PDF splitting API in C# with the help of PDF.co Web API. For implimentation of this functionality, please copy and paste code below into your app using code editor. Then compile and run your app. Tutorials are available along with installed PDF.co Web API if you'd like to dive deeper into the topic and the details of the API.

Our website provides free trial version of PDF.co Web API that includes source code samples to help with your C# project.

C# - Program.cs

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

// Cloud API asynchronous "Split 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 = "*****";

        // Source PDF file to split
        const string SourceFileUrl = @"https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-split/sample.pdf";
        // Comma-separated list of page numbers (or ranges) to process.
        // Example: '1,3-5,7-'
    }
}
```

```

const string Pages = "1-2,3-";
// (!) 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);

    try
    {
        // Prepare URL for `Split PDF` API call
        string query = Uri.EscapeUriString(string.Format(
            "https://api.pdf.co/v1/pdf/split?pages=
{0}&url={1}&async={2}",
            Pages,
            SourceFileUrl,
            Async));

        // 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 JSON file available after
            // the job completion; it will contain URLs of result PDF files.
            string resultJsonFileUrl =
            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
            // checking and completion.
            // to use a separate thread for the status
            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 JSON file as
                    string jsonFileString =
                    webClient.DownloadString(resultJsonFileUrl);

```

```

JArray.Parse(jsonFileString);

files
resultFilesUrls)

= token.ToString();
= String.Format(@"\part{0}.pdf", part);

webClient.DownloadFile(resultFileUrl, localFileName);

Console.WriteLine("Downloaded \"{0}\".", localFileName);

JArray resultFilesUrls =
// Download generated PDF
int part = 1;
foreach (JToken token in
{
    string resultFileUrl
    string localFileName

    part++;
}
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);

        + jobId;

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

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

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

C# - 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](#)