

www.bytescout.com

How to convert PDF to JSON from uploaded file asynchronously for PDF to JSON API in C# using ByteScout Cloud API Server

Step By Step Instructions on how to convert PDF to JSON from uploaded file asynchronously for PDF to JSON API in C#

These source code samples are listed and grouped by their programming language and functions they use. ByteScout Cloud API Server helps with PDF to JSON API in C#. 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 connection 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!.

This simple and easy to understand sample source code in C# for ByteScout Cloud API Server contains different functions and options you should do calling the API to implement PDF to JSON API. For implementation of this functionality, please copy and paste the code below into your app using code editor. Then compile and run your app. Writing C# application mostly includes various stages of the software development so even if the functionality works please check it with your data and the production environment.

Our website provides free trial version of ByteScout Cloud API Server that gives source code samples to assist with your C# project.

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.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", "ByteScoutWebApiExample", {A8A8A8A8-A8A8-A8A8-A8A8-A8A8A8A8A8A8}
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|Any CPU
        {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
EndGlobal
```

Program.cs

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

// Please NOTE: In this sample we're assuming Cloud API Server is hosted at "https://localhost:9000"
// If it's not then please replace this with your hosting url.

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

namespace ByteScoutWebApiExample
{
    class Program
    {
        // Source PDF file
        const string SourceFile = @"..\sample.pdf";
    }
}
```

```
// Comma-separated list of page indices (or ranges) to process. Leave empty for all pages.
const string Pages = "";

// PDF document password. Leave empty for unprotected documents.
const string Password = "";
// Destination JSON file name
const string DestinationFile = @"..\result.json";
// (!) Make asynchronous job
const bool Async = true;

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

    // Upload file
    string uploadedImageUrl = UploadFile(webClient, SourceFile);

    // Prepare URL for `PDF To JSON` API call
    string query = Uri.EscapeUriString(string.Format(
        "https://localhost/pdf/convert/to/json?name={0}&password={1}&pages={2}",
        Path.GetFileName(DestinationFile),
        Password,
        Pages,
        uploadedImageUrl,
        Async));

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

        // 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 JSON file that will available once the job is completed
            string resultImageUrl = json["url"].ToString();

            // Check the job status in a loop.
            // If you don't want to pause the main thread you can
            // to use a separate thread for the status check
            do
            {
                string status = CheckJobStatus(webClient, jobId);

                // Display timestamp and status (for debugging purposes)
                Console.WriteLine(DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss") + " - " + status);

                if (status == "success")
                {
                    // Download JSON file
                    webClient.DownloadFile(resultImageUrl, DestinationFile);
                    // Display success message
                    Console.WriteLine("Generated JSON file successfully!");
                    break;
                }
                else if (status == "working")
                {
                    // Display working message
                    Console.WriteLine("Job is still working...");
                }
            } while (status != "success");
        }
    }
}
```

```

        {
            // 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 UploadFile(WebClient webClient, string file)
{
    // RETRIEVE THE PRESIGNED URL TO UPLOAD THE FILE:

    // Prepare URL for `Get Presigned URL` API call
    string query = Uri.EscapeUriString(string.Format(
        "https://localhost/file/upload/get-presigned-url?contenttype=application/" +
        Path.GetFileName(SourceFile)));
}

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

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

string uploadUrl = json["presignedUrl"].ToString();
string uploadedFileUrl = json["url"].ToString();

// UPLOAD FILE:

webClient.Headers.Add("content-type", "application/octet-stream");
webClient.UploadFile(uploadUrl, "PUT", SourceFile); // You can use UploadData
webClient.Headers.Remove("content-type");

return uploadedFileUrl;
}

static string CheckJobStatus(WebClient webClient, string jobId)
{
    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="12.0.1" 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](http://www.ByteScout.com)

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

www.bytescout.com