

How to convert DOC to PDF from URL asynchronously for DOC to PDF API in C# and PDF.co Web API

Learn how to convert DOC to PDF from URL asynchronously to have DOC to PDF API in C#

Source code documentation samples provide quick and easy way to add a required functionality into your application. PDF.co Web API was made to help with DOC to PDF 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.

C# code snippet like this for PDF.co Web API works best when you need to quickly implement DOC to PDF API in your C# application. This C# sample code should be copied and pasted into your project. After doing this just compile your project and click Run. Enjoy writing a code with ready-to-use sample C# codes to add DOC to PDF API functions using PDF.co Web API in C#.

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 System.Threading;
using Newtonsoft.Json.Linq;

// Cloud API asynchronous "DOC To 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 URL of source DOC or DOCX file.
        const string SourceFileUrl = "https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/doc-to-pdf/sample.docx";
        // 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 `DOC To PDF` API call
    string query = Uri.EscapeUriString(string.Format(
        "{0}&url={1}&async={2}",
        Path.GetFileName(DestinationFile),
        SourceFileUrl,
        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
            // be 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
            // checking and completion.
            // 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"]);
    }
}
}
}
}

```



FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about PDF.co Web API](#)

[Explore documentation](#)

[Visit \[www.ByteScout.com\]\(http://www.ByteScout.com\)](#)

or

[Get Your Free API Key for \[www.PDF.co\]\(http://www.PDF.co\) Web API](#)