

How to convert PDF to PNG from URL asynchronously for PDF to image API in C# with PDF.co Web API

How to convert PDF to PNG from URL asynchronously in C# with easy ByteScout code samples to make PDF to image API. Step-by-step tutorial

The documentation is designed to help you to implement the features on your side. PDF to image API in C# can be implemented with PDF.co Web API. 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.

The SDK samples like this one below explain how to quickly make your application do PDF to image API in C# with the help of PDF.co Web API. This C# sample code should be copied and pasted into your project. After doing this just compile your project and click Run. Further enhancement of the code will make it more vigorous.

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 "PDF To PNG" 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
        const string SourceFileUrl = @"https://bytescout-
com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-to-image/sample.pdf";
        // Comma-separated list of page indices (or ranges) to process. Leave
```

empty for all pages. Example: '0,2-5,7-'.
const string Pages = "";

// PDF document password. Leave empty for unprotected documents.
const string Password = "";

// (!) 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 `PDF To PNG` API call
        string query = Uri.EscapeUriString(string.Format(
            "https://api.pdf.co/v1/pdf/convert/to/png?
password={0}&pages={1}&url={2}&async={3}",
            Password,
            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 PNG 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

```
            // 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")
                {
```

```

string // Download JSON file as
webClient.DownloadString(resultJsonFileUrl);
string jsonFileString =
JSONArray.Parse(jsonFileString);
JSONArray resultFilesUrls =
files // Download generated PNG
resultFilesUrls
int page = 1;
foreach (JToken token in
{
string resultFileUrl
string localFileName
= token.ToString();
= String.Format(@".\page{0}.png", page);
webClient.DownloadFile(resultFileUrl, localFileName);
Console.WriteLine("Downloaded \"{0}\".", localFileName);
page++;
}
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"]);
    }
}
}
```

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](#)