

[www.bytescout.com](http://www.bytescout.com)

## How to convert PDF to XLS from URL asynchronously for PDF to excel API in C# and ByteScout Cloud API Server

### How to convert PDF to XLS from URL asynchronously for PDF to excel API in C#: Step By Step Instructions

The sample source codes on this page will show you how to create PDF to excel API in C#. ByteScout Cloud API Server was designed to assist PDF to excel API in C#. ByteScout Cloud API Server is the ready to use Web API Server that can be deployed in less than 30 minutes into your own in-house server 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 excel API. Open your C# project and simply copy & paste the code and then run your app! Easy to understand tutorials are available along with installed ByteScout Cloud API Server if you'd like to learn more about the topic and the details of the API.

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:3001"
// If it's not then please replace this with your hosting url.

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

namespace ByteScoutWebApiExample
{
    class Program
    {
        // Direct URL of source PDF file.
        const string SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/f.../source.pdf";
        // Direct URL of output XLS file.
        const string DestinationFileUrl = "https://bytescout-com.s3.amazonaws.com/f.../output.xls";
    }
}
```

```

// Comma-separated list of page indices (or ranges) to process. Leave empty to process all pages.
const string Pages = "";
// PDF document password. Leave empty for unprotected documents.
const string Password = "";
// Destination XLS file name
const string DestinationFile = @".\result.xls";
// (!) Make asynchronous job
const bool Async = true;

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

    // Prepare URL for `PDF To XLS` API call
    string query = Uri.EscapeUriString(string.Format(
        "https://localhost/pdf/convert/to/xls?name={0}&password={1}&{2}&{3}&{4}&{5}&{6}&{7}",
        Path.GetFileName(DestinationFile),
        Password,
        Pages,
        SourceFilePath,
        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 XLS file that will available once the job is completed
        string resultFileUrl = 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(jobId);

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

            if (status == "success")
            {
                // Download XLS file
                webClient.DownloadFile(resultFileUrl, "result.xls");
                // Print message to console
                Console.WriteLine("Generated XLS file successfully!");
                break;
            }
            else if (status == "working")
            {
                // Pause for a few seconds
                Thread.Sleep(10000);
            }
        } while (status != "success");
    }
}

```

```

        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())
    {
        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="10.0.3" 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](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)