

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

## PDF fill PDF forms in C# and PDF.co Web API

PDF.co Web API: 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.

**FOR MORE INFORMATION AND FREE TRIAL:**

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

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

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

## Source Code Files:

**FillPDFForms.csproj**

```
<?xml version="1.0" encoding="utf-8"?>  
<Project ToolsVersion="12.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.co
```

```

<Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.pr
<PropertyGroup>
  <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
  <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
  <ProjectGuid>{1E1C2C34-017E-4605-AE2B-55EA3313BE51}</ProjectGuid>
  <OutputType>Exe</OutputType>
  <RootNamespace>FillPDFForms</RootNamespace>
  <AssemblyName>FillPDFForms</AssemblyName>
  <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
  <FileAlignment>512</FileAlignment>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugSymbols>>true</DebugSymbols>
  <DebugType>full</DebugType>
  <Optimize>>false</Optimize>
  <OutputPath>bin\Debug\</OutputPath>
  <DefineConstants>DEBUG;TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugType>pdbonly</DebugType>
  <Optimize>>true</Optimize>
  <OutputPath>bin\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
</PropertyGroup>
<ItemGroup>
  <Reference Include="Newtonsoft.Json, Version=10.0.0.0, Culture=neutral, PublicKeyT
    <HintPath>packages\Newtonsoft.Json.10.0.3\lib\net40\Newtonsoft.Json.dll</HintPat
    <Private>True</Private>
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data" />
  <Reference Include="System.Xml" />
</ItemGroup>
<ItemGroup>
  <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
  <None Include="packages.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
  Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

## FillPDFForms.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.26730.10
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "FillPDFForms", "FillPDFForms.csproj"
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
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {238BD6FC-F70A-4B5C-B639-34E5B171A981}
    EndGlobalSection
EndGlobal
```

## Program.cs

```
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using System;
using System.Collections.Generic;
using System.Net;
using System.Runtime.InteropServices;
using System.Threading;

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 PDF file.
const string SourceFileUrl = "https://bytescout-com.s3-us-west-2.amazonaws.com";
// PDF document password. Leave empty for unprotected documents.
const string Password = "";
// File name for generated output. Must be a String
const string FileName = "f1040-form-filled";

// Destination File Name
const string DestinationFile = "./result.pdf";

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);

    // Values to fill out pdf fields with built-in pdf form filler
    var fields = new List<object> {
        new { fieldName = "topmostSubform[0].Page1[0].FilingStatus[0].c1_01[1]"},
        new { fieldName = "topmostSubform[0].Page1[0].f1_02[0]", pages = "1"},
        new { fieldName = "topmostSubform[0].Page1[0].f1_03[0]", pages = "1"},
        new { fieldName = "topmostSubform[0].Page1[0].YourSocial_ReadOrderCont"},
        new { fieldName = "topmostSubform[0].Page1[0].YourSocial_ReadOrderCont"},
        new { fieldName = "topmostSubform[0].Page1[0].YourSocial_ReadOrderCont"},
        new { fieldName = "topmostSubform[0].Page1[0].YourSocial_ReadOrderCont"}
    };

    // If enabled, Runs processing asynchronously. Returns Use JobId that you
    var async = true; // (!) Make asynchronous job

    // Prepare requests params as JSON
    // See documentation: https://apidocs.pdf.co
    Dictionary<string, object> parameters = new Dictionary<string, object>();
    parameters.Add("url", SourceFileUrl);
    parameters.Add("name", FileName);
    parameters.Add("password", Password);
    parameters.Add("async", async);
    parameters.Add("fields", fields);

    // Convert dictionary of params to JSON
    string jsonPayload = JsonConvert.SerializeObject(parameters);

    try
    {
        // URL of "PDF Edit" endpoint
        string url = "https://api.pdf.co/v1/pdf/edit/add";

        // Execute POST request with JSON payload
        string response = webClient.UploadString(url, jsonPayload);

        // 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 PDF file that will available after the job com
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 c
// to use a separate thread for the status checking and completion
do
{
    string status = CheckJobStatus(jobId); // Possible statuses: "

    // Display timestamp and status (for demo purposes)
    Console.WriteLine(DateTime.Now.ToLongTimeString() + ": " + sta

    if (status == "success")
    {
        // Download PDF file
        webClient.DownloadFile(resultFileUrl, DestinationFile);

        Console.WriteLine("Generated PDF file saved as \"{0}\" fil
        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());
}
finally
{
    webClient.Dispose();
}

Console.WriteLine();
Console.WriteLine("Press any key...");
Console.ReadKey();
}

/// <summary>
/// Check job status
/// </summary>
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"]);
}
}
```

## 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 PDF.co Web API Home Page](#)  
[Explore PDF.co Web API Documentation](#)

[Explore Samples](#)  
[Sign Up for PDF.co Web API 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](http://www.bytescout.com)