

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.com
```

```

<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;

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
    // To fill fields in PDF form, use the following format page;f
    // To fill checkbox, use true, for example: 0;checkbox1>true.
    // To separate multiple objects, use | separator.
    // To get the list of all fillable fields in PDF form please u
    var fieldString = "1;topmostSubform[0].Page1[0].f1_02[0];John .

    // If enabled, Runs processing asynchronously. Returns Use Job
    var async = false;

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

// 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)
        {
            // Get URL of generated PDF file
            string resultFileUrl = json["url"].ToString();

            // Download generated PDF file

```

```

        webClient.DownloadFile(resultFileUrl, Destinatio
        Console.WriteLine("Generated PDF file saved as
    }
    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();
}
}
}

```

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