

www.bytescout.com

How to add text and images to PDF in C# and ByteScout Cloud API Server

This code in C# shows how to add text and images to PDF with this how to tutorial

The coding instructions are formulated to help you to try-out the features without the requirement to write your own code. Want to add text and images to PDF in your C# app? ByteScout Cloud API Server is designed for it. ByteScout Cloud API Server is API server that is ready to use and can be installed and deployed in less than 30 minutes on your own Windows server or server in a cloud. It can save data and files on your local server-based file storage or in Amazon AWS S3 storage. Data is processed solely on the API server and is powered by ByteScout engine, no cloud services or Internet connection is required for data processing..

This prolific sample source code in C# for ByteScout Cloud API Server contains various functions and other necessary options you should do calling the API to add text and images to PDF. IF you want to implement the functionality, just copy and paste this code for C# below into your code editor with your app, compile and run your application. Check C# sample code samples to see if they respond to your needs and requirements for the project.

The trial version of ByteScout Cloud API Server can be downloaded for free from our website. It also includes source code samples for C# and other programming languages.

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:

AddTextByFindingTargetCoordinates.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}") = "AddTextByFindingTargetCoordinates"
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
        {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Release|Any CPU.ActiveCfg = Release
        {1E1C2C34-017E-4605-AE2B-55EA3313BE51}.Release|Any CPU.Build.0 = Release
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {238BD6FC-F70A-4B5C-B639-34E5B171A981}
    EndGlobalSection
EndGlobal
```

Program.cs

```
using System;
using System.IO;
using System.Net;
using Newtonsoft.Json.Linq;

namespace ByteScoutWebApiExample
{
    // Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://bytescout-com.s3.amazonaws.com/files/demo-files/test.pdf"
    // If it's not then please replace this with your hosting url.

    class Program
    {
        // Direct URL of source PDF file.
        const string SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/test.pdf";
        // Comma-separated list of page indices (or ranges) to process. Leave empty for all pages.
        const string Pages = "1-3";
        // Path to output PDF file
        const string OutputFileName = "output.pdf";
    }
}
```

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

// Destination PDF file name
const string DestinationFile = @".\result.pdf";

// Annotation params
private const string Type = "annotation";
private const string Text = "Some notes will go here... Some notes will go here";
private const string FontName = "Times New Roman";
private const float FontSize = 12;
private const string Color = "FF0000";

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

    // * Add text annotation *

    // Find Text coordinates to add Annotation
    var oCoordinates = FindCoordinates(SourceFileUrl, "Notes");

    var X = oCoordinates.x;
    var Y = oCoordinates.y + 25;

    // Prepare URL for `PDF Edit` API call
    string query = Uri.EscapeUriString(string.Format(
        "https://localhost/pdf/edit/add?name={0}&password={1}&pages={2}&url={3}",
        Path.GetFileName(DestinationFile),
        Password,
        Pages,
        SourceFileUrl,
        Type,
        X,
        Y,
        Text,
        FontName,
        FontSize,
        Color));

    try
    {
        // Execute request
        string response = webClient.DownloadString(query);

        // 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 PDF file
            webClient.DownloadFile(resultFileUrl, DestinationFile);

            Console.WriteLine("Generated PDF file saved as \"{}\" file.", Desti

```

```

        }
    else
    {
        Console.WriteLine(json["message"].ToString());
    }
}

catch (WebException e)
{
    Console.WriteLine(e.ToString());
}

webClient.Dispose();

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

/// <summary>
/// Find result coordinates
/// </summary>
public static ResultCoOrdinates FindCoordinates(string SourceFileUrl, string SearchString)
{
    ResultCoOrdinates oResult = null;

    // Create standard .NET web client instance
    WebClient webClient = new WebClient();

    // Prepare URL for PDF text search API call.
    string query = Uri.EscapeUriString(string.Format(
        "https://localhost/pdf/find?url={0}&searchString={1}",
        SourceFileUrl,
        SearchString));

    try
    {
        // Execute request
        string response = webClient.DownloadString(query);

        // Parse JSON response
        JObject json = JObject.Parse(response);

        if (json["status"].ToString() != "error")
        {
            JToken item = json["body"][0];

            oResult = new ResultCoOrdinates
            {
                x = Convert.ToInt32(item["left"]),
                y = Convert.ToInt32(item["top"]),
                width = Convert.ToInt32(item["width"]),
                height = Convert.ToInt32(item["height"])
            };
        }
    catch (WebException e)
    {
        Console.WriteLine(e.ToString());
    }
}

```

```
        webClient.Dispose();

        return oResult;
    }

    public class ResultCoordinates
{
    public int x { get; set; }
    public int y { get; set; }
    public int width { get; set; }
    public int height { get; set; }
}

}
```

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](#)