

How to convert pdf to html with images with pdf to html sdk in ASP.NET and ByteScout PDF Suite

Learn to convert pdf to html with images with pdf to html sdk in ASP.NET

The documentation is designed for a specific purpose to help you to apply the features on your side. ByteScout PDF Suite: the set that includes 6 SDK products to work with PDF from generating rich PDF reports to extracting data from PDF documents and converting them to HTML. This bundle includes PDF (Generator) SDK, PDF Renderer SDK, PDF Extractor SDK, PDF to HTML SDK, PDF Viewer SDK and PDF Generator SDK for Javascript. It can convert pdf to html with images with pdf to html sdk in ASP.NET.

Want to save time? You will save a lot of time on writing and testing code as you may just take the ASP.NET code from ByteScout PDF Suite for convert pdf to html with images with pdf to html sdk below and use it in your application. IF you want to implement the functionality, just copy and paste this code for ASP.NET below into your code editor with your app, compile and run your application. This basic programming language sample code for ASP.NET will do the whole work for you to convert pdf to html with images with pdf to html sdk.

ByteScout provides the free trial version of ByteScout PDF Suite along with the documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF Suite](#)

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

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

Source Code Files:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="Ex
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xl

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>

    </div>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using Bytescout.PDF2HTML;

namespace ExtractHTML
{
    public partial class _Default : System.Web.UI.Page
    {
        /*
        IF YOU SEE TEMPORARY FOLDER ACCESS ERRORS:

        Temporary folder access is required for web application when you use By
        If you are getting errors related to the access to temporary folder lil

        SOLUTION:

        If your IIS Application Pool has "Load User Profile" option enabled the
        If you are running Web Application under an impersonated account or IIS

        In this case
        - check the User or User Group your web application is running under
        - then add permissions for this User or User Group to read and write in
        - restart your web application and try again
```

```

*/

protected void Page_Load(object sender, EventArgs e)
{
    // This test file will be copied to the project directory on the
    String inputFile = Server.MapPath(@"..\bin\sample2.pdf");

    // Create Bytescout.PDF2HTML.HTMLExtractor instance
    HTMLExtractor extractor = new HTMLExtractor();
    extractor.RegistrationName = "demo";
    extractor.RegistrationKey = "demo";

    // Set HTML with CSS extraction mode
    extractor.ExtractionMode = HTMLExtractionMode.HTMLWithCSS;

    // Embed images into HTML file
    extractor.SaveImages = ImageHandling.Embed;

    // Load sample PDF document
    extractor.LoadDocumentFromFile(inputFile);

    Response.Clear();
    Response.ContentType = "text/html";

    // Save extracted text to output stream
    extractor.SaveHtmlToStream(Response.OutputStream);

    Response.End();

    extractor.Dispose();
}
}
}

```

Default.aspx.designer.cs

```

//-----
// <auto-generated>
//   This code was generated by a tool.
//   Runtime Version:2.0.50727.4952
//
//   Changes to this file may cause incorrect behavior and will be lost if
//   the code is regenerated.
// </auto-generated>
//-----

namespace ExtractHTML
{

```

```

/// <summary>
/// _Default class.
/// </summary>
/// <remarks>
/// Auto-generated class.
/// </remarks>
public partial class _Default {

    /// <summary>
    /// form1 control.
    /// </summary>
    /// <remarks>
    /// Auto-generated field.
    /// To modify move field declaration from designer file to code-behind file.
    /// </remarks>
    protected global::System.Web.UI.HtmlControls.HtmlForm form1;
}
}

```

ExtractHTML.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}") = "ExtractHTML", "ExtractHTML.csproj"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Debug|Any CPU.ActiveCfg = Debug
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Debug|Any CPU.Build.0 = Debug
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Release|Any CPU.ActiveCfg = Release
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Release|Any CPU.Build.0 = Release
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
EndGlobal

```

Web.config

```
<?xml version="1.0"?>

<configuration>

  <appSettings/>
  <connectionStrings/>

  <system.web>
    <!--
      Set compilation debug="true" to insert debugging
      symbols into the compiled page. Because this
      affects performance, set this value to true only
      during development.
    -->
    <compilation debug="true" />
    <!--
      The <authentication> section enables configuration
      of the security authentication mode used by
      ASP.NET to identify an incoming user.
    -->
    <authentication mode="Windows" />
    <!--
      The <customErrors> section enables configuration
      of what to do if/when an unhandled error occurs
      during the execution of a request. Specifically,
      it enables developers to configure html error pages
      to be displayed in place of a error stack trace.

      <customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">
        <error statusCode="403" redirect="NoAccess.htm" />
        <error statusCode="404" redirect="FileNotFound.htm" />
      </customErrors>
    -->
  </system.web>
</configuration>
```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout PDF Suite Home Page](#)
[Explore ByteScout PDF Suite Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout PDF Suite 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