

How to get PDF document information in ASP.NET C# with ByteScout PDF Extractor SDK

Write code in ASP.NET C# to get PDF document information with this step-by-step tutorial

The sample source codes on this page shows how to get PDF document information in ASP.NET C#. ByteScout PDF Extractor SDK can get PDF document information. It can be used from ASP.NET C#. ByteScout PDF Extractor SDK is the SDK is designed to help developers with pdf tables and pdf data extraction from unstructured documents like pdf, tiff, scans, images, scanned and electronic forms. The library is powered by OCR, computer vision and AI to provide unique functionality like table detection, automatic table structure extraction, data restoration, data restructuring and reconstruction. Supports PDF, TIFF, PNG, JPG images as input and can output CSV, XML, JSON formatted data. Includes full set of utilities like pdf splitter, pdf merger, searchable pdf maker and other utilities.

Fast application programming interfaces of ByteScout PDF Extractor SDK for ASP.NET C# plus the instruction and the code below will help you quickly learn how to get PDF document information. This ASP.NET C# sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Test ASP.NET C# sample code examples whether they respond your needs and requirements for the project.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Extractor SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF Extractor SDK](#)

[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.PDFExtractor;

namespace ExtractInfo
{
    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\sample1.pdf");

    // Create Bytescout.PDFExtractor.InfoExtractor instance
    InfoExtractor extractor = new InfoExtractor();
    extractor.RegistrationName = "demo";
    extractor.RegistrationKey = "demo";

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

    // Display document information

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

    Response.Write("Author: " + extractor.Author + "<br/>");
    Response.Write("Creator: " + extractor.Creator + "<br/>");
    Response.Write("Producer: " + extractor.Producer + "<br/>");
    Response.Write("Subject: " + extractor.Subject + "<br/>");
    Response.Write("Title: " + extractor.Title + "<br/>");
    Response.Write("CreationDate: " + extractor.CreationDate + "<br/>");
    Response.Write("Keywords: " + extractor.Keywords + "<br/>");
    Response.Write("Bookmarks: " + extractor.Bookmarks + "<br/>");
    Response.Write("Encrypted: " + extractor.Encrypted + "<br/>");

    Response.End();
}
}
}

```

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 ExtractInfo {

```

```

/// <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;
}
}

```

ExtractInfo.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}") = "ExtractInfo", "ExtractInfo.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

```

```
<?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=s28W3_KMraU

ON-PREMISE OFFLINE SDK

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