reading and writing to azure blob in C# and ByteScout PDF Extractor SDK

Simple tutorial on how to do reading and writing to azure blob in C#

Easy to understand coding instructions are written to assist you to try-out the features without the requirement to write your own code. ByteScout PDF Extractor SDK helps with reading and writing to azure blob in 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.

The below SDK samples describe how to quickly make your application do reading and writing to azure blob in C# with the help of ByteScout PDF Extractor SDK. Follow the steps-by-step instructions from the scratch to work and copy and paste code for C# into your editor. Enjoy writing a code with ready-to-use sample C# codes to add reading and writing to azure blob functions using ByteScout PDF Extractor SDK in C#.

ByteScout PDF Extractor SDK free trial version is available for download from our website. Free trial also includes 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

Source Code Files:

Program.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Diagnostics;
using Bytescout.PDFExtractor;
using System.IO;

namespace ConsoleApplication
{
    class Program
    {
        static void Main(string[] args)
        {

            /*
             Please note: Streams can be read/write to azure blobs, so in this example
             we're demonstrating how to read pdf from stream, convert to csv,
             and write to csv stream
             */

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

            // Get input stream
            var inputStream = GetMemoryStream("sample3.pdf");

            // Load sample PDF document from stream
            extractor.LoadDocumentFromStream(inputStream);

            // you can change CSV separator symbol (if needed) from "," symbol to anot
            //extractor.CSVSeparatorSymbol = ",";

            // Save extracted CSV data to output stream
            var outputStream = new MemoryStream();
            extractor.SaveCSVToStream(outputStream);

            // Save output stream to file, so we can take a look
            WriteStreamToFile(outputStream, "output.csv");

            // Cleanup
            extractor.Dispose();

            Console.WriteLine();
            Console.WriteLine("Data has been extracted to 'output.csv' file.");
            Console.WriteLine();
            Console.WriteLine("Press any key to continue and open CSV in default CSV v
            Console.ReadKey();

            // Open result document in default associated application (for demo purpos
            ProcessStartInfo processStartInfo = new ProcessStartInfo("output.csv");
            processStartInfo.UseShellExecute = true;
            Process.Start(processStartInfo);
        }

        /// <summary>
        /// Get memory stream from file
        /// </summary>
        public static Stream GetMemoryStream(string fileName)
        {
            return new FileStream(fileName, FileMode.Open);
        }

        /// <summary>
        /// Write stream to file
        /// </summary>
        public static void WriteStreamToFile(Stream stream, string fileName)
        {
            using (var fileStream = File.Create(fileName))
            {
                stream.Seek(0, SeekOrigin.Begin);
                stream.CopyTo(fileStream);
            }
        }

    }
}
```

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