reading and writing to azure blob in VB.NET and ByteScout PDF Extractor SDK

reading and writing to azure blob in VB.NET

An easy to understand guide to learn how to reading and writing to azure blob in VB.NET. ByteScout PDF Extractor SDK helps with reading and writing to azure blob in VB.NET. ByteScout PDF Extractor SDK is the Software Development Kit (SDK) that is designed to help developers with 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.

This rich and prolific sample source code in VB.NET for ByteScout PDF Extractor SDK contains various functions and options you should do calling the API to implement reading and writing to azure blob. If you want to implement this functionality, you should copy and paste code below into your app using code editor. Then compile and run your application. Updated and detailed documentation and tutorials are available along with installed ByteScout PDF Extractor SDK if you'd like to learn more about the topic and the details of the API.

Trial version can be downloaded from our website for free. It contains this and other source code samples for VB.NET.

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:

Module1.vb

```vb
Imports System
Imports System.Collections.Generic
Imports System.Text
Imports Bytescout.PDFExtractor
Imports System.Diagnostics
Imports System.IO

Namespace ConsoleApplication1

    Class Program

        Shared Sub Main(ByVal args As String())

            '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
            Dim extractor As New CSVExtractor()
            extractor.RegistrationName = "demo"
            extractor.RegistrationKey = "demo"

            ' Get input stream
            Dim inputStream = GetMemoryStream("sample3.pdf")

            ' Load sample PDF document from Stream
            extractor.LoadDocumentFromStream(inputStream)

            ' You can change CSV separator symbol (if needed) from "," symbol to anothe
            'extractor.CSVSeparatorSymbol = ","

            ' Save extracted CSV data to output stream
            Dim outputStream As 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 file in default associated application (for demo purposes)
            Process.Start("output.csv")
```

```vb
        End Sub

        ''' <summary>
        ''' Get Memory stream from file
        ''' </summary>
        Shared Function GetMemoryStream(ByVal fileName As String) As Stream
            Return New FileStream(fileName, FileMode.Open)
        End Function

        ''' <summary>
        ''' Write stream to file
        ''' </summary>
        Shared Sub WriteStreamToFile(ByVal stream As Stream, ByVal fileName As String)

            Using fileStream = File.Create(fileName)
                stream.Seek(0, SeekOrigin.Begin)
                stream.CopyTo(fileStream)
            End Using

        End Sub

    End Class
End Namespace
```

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](www.ByteScout.com)

[visit www.PDF.co](www.PDF.co)