www.bytescout.com

How to find table in pdf and extract as xml with pdf extractor sdk in VB.NET with ByteScout Premium Suite

How to write a robust code in VB.NET to find table in pdf and extract as xml with pdf extractor sdk with this step-by-step tutorial

An easy to understand guide on how to find table in pdf and extract as xml with pdf extractor sdk in VB.NET with this source code sample. ByteScout Premium Suite: the bundle that includes twelve SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording. It can find table in pdf and extract as xml with pdf extractor sdk in VB.NET.

These VB.NET code samples for VB.NET guide developers to speed up coding of the application when using ByteScout Premium Suite. Simply copy and paste in your VB.NET project or application you and then run your app! If you want to use these VB.NET sample examples in one or many applications then they can be used easily.

Our website gives trial version of ByteScout Premium Suite for free. It also includes documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Premium Suite

Explore API Documentation

Get Free Training for ByteScout Premium Suite

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Program.vb

```
Imports Bytescout.PDFExtractor

Class Program
        Friend Shared Sub Main(args As String())

        ' Create Bytescout.PDFExtractor.XMLExtractor instance
        Dim xmlExtractor As New XMLExtractor()
        xmlExtractor.RegistrationName = "demo"
        xmlExtractor.RegistrationKey = "demo"

        ' Create Bytescout.PDFExtractor.TableDetector instance
        Dim tableDetector As New TableDetector()
        tableDetector.RegistrationName = "demo"
        tableDetector.RegistrationKey = "demo"

        ' We should define what kind of tables we should detect.
        ' So we set min required number of columns to 3 ...
        tableDetector.DetectionMinNumberOfColumns = 3
        ' ... and we set min required number of rows to 3
        tableDetector.DetectionMinNumberOfRows = 3

                ' Load sample PDF document
        xmlExtractor.LoadDocumentFromFile(".\sample3.pdf")
        tableDetector.LoadDocumentFromFile(".\sample3.pdf")

                ' Get page count
        Dim pageCount As Integer = tableDetector.GetPageCount()

                For i As Integer = 0 To pageCount - 1
            Dim t As Integer = 1
            ' Find first table and continue if found
            If (tableDetector.FindTable(i)) Then
                Do
                    ' Set extraction area for XML extractor to rectangle received from
                    xmlExtractor.SetExtractionArea(tableDetector.FoundTableLocation)
                    ' Export the table to XML file
                    xmlExtractor.SavePageXMLToFile(i, "page-" + i.ToString() + "-table-
                    t = t + 1
                Loop While tableDetector.FindNextTable()
            End If
        Next

        ' Cleanup
                xmlExtractor.Dispose()
                tableDetector.Dispose()

        ' Open first output file in default associated application (for demo purposes)
        System.Diagnostics.Process.Start("page-0-table-1.xml")

        End Sub
End Class
```

VIDEO

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

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Premium Suite Home Page
Explore ByteScout Premium Suite Documentation
Explore Samples
Sign Up for ByteScout Premium Suite 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