www.bytescout.com

How to parse with OCR for document parser API in VB.NET using ByteScout Cloud API Server

ByteScout Cloud API Server: the ready to use Web API Server that can be deployed in less than 30 minutes into your own in-house server or into private cloud server. Can store data on in-house local server based storage or in Amazon AWS S3 bucket. Processing data solely on the server using buil-in ByteScout powered engine, no cloud services are used to process your data!.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about **ByteScout Cloud API Server**

Explore API Documentation

Get Free Training for ByteScout Cloud API Server

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

ByteScoutWebApiExample.sln

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.26730.10
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "ByteScoutWebApiExample", "ByteSco
EndProject
Global
        GlobalSection(SolutionConfigurationPlatforms) = preSolution
                Debug|Any CPU = Debug|Any CPU
                Release|Any CPU = Release|Any CPU
        EndGlobalSection
        GlobalSection(ProjectConfigurationPlatforms) = postSolution
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.ActiveCfg = Debug
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.Build.0 = Debug|A
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.ActiveCfg = Rel
                {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.Build.0 = Relea
        EndGlobalSection
        GlobalSection(SolutionProperties) = preSolution
                HideSolutionNode = FALSE
        EndGlobalSection
        GlobalSection(ExtensibilityGlobals) = postSolution
                SolutionGuid = {4576C9BB-A42D-46A8-9198-7E2982E122FA}
        EndGlobalSection
EndGlobal
```

ByteScoutWebApiExample.vbproj

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/200
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.pr
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>ByteScoutWebApiExample.Module1</StartupObject>
    <RootNamespace>ByteScoutWebApiExample</RootNamespace>
    <AssemblyName>ByteScoutWebApiExample</AssemblyName>
    <FileAlignment>512</FileAlignment>
    <MyType>Console</MyType>
    <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugSymbols>true</DebugSymbols>
    <DebugType>full</DebugType>
    <DefineDebug>true</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <OutputPath>bin\Debug\</OutputPath>
```

```
    <DocumentationFile>ByteScoutWebApiExample.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
    <PlatformTarget>AnyCPU</PlatformTarget>
    <DebugType>pdbonly</DebugType>
    <DefineDebug>false</DefineDebug>
    <DefineTrace>true</DefineTrace>
    <Optimize>true</Optimize>
    <OutputPath>bin\Release\</OutputPath>
    <DocumentationFile>ByteScoutWebApiExample.xml</DocumentationFile>
    <NoWarn>42016,41999,42017,42018,42019,42032,42036,42020,42021,42022</NoWarn>
  </PropertyGroup>
  <PropertyGroup>
    <OptionExplicit>On</OptionExplicit>
  </PropertyGroup>
  <PropertyGroup>
    <OptionCompare>Binary</OptionCompare>
  </PropertyGroup>
  <PropertyGroup>
    <OptionStrict>Off</OptionStrict>
  </PropertyGroup>
  <PropertyGroup>
    <OptionInfer>On</OptionInfer>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="Newtonsoft.Json, Version=10.0.0.0, Culture=neutral, PublicKeyT
      <HintPath>packages\Newtonsoft.Json.10.0.3\lib\net40\Newtonsoft.Json.dll</HintPat
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
    <Reference Include="System.Xml" />
    <Reference Include="System.Core" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
  </ItemGroup>
  <ItemGroup>
    <Import Include="Microsoft.VisualBasic" />
    <Import Include="System" />
    <Import Include="System.Collections" />
    <Import Include="System.Collections.Generic" />
    <Import Include="System.Data" />
    <Import Include="System.Diagnostics" />
    <Import Include="System.Linq" />
    <Import Include="System.Xml.Linq" />
  </ItemGroup>
  <ItemGroup>
    <Compile Include="Module1.vb" />
  </ItemGroup>
  <ItemGroup>
    <None Include="packages.config" />
    <None Include="DigitalOcean-scanned.jpg">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </None>
    <None Include="DigitalOcean.yml">
      <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </None>
  </ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
```

```
</Project>
```

DigitalOcean.yml

```
templateName: DigitalOcean Invoice
templateVersion: 4
templatePriority: 0
detectionRules:
  keywords:
  - DigitalOcean
  - 101 Avenue of the Americas
  - Invoice Number
objects:
- name: companyName
  objectType: field
  fieldProperties:
    fieldType: static
    expression: DigitalOcean
    regex: true
- name: invoiceId
  objectType: field
  fieldProperties:
    fieldType: macros
    expression: 'Invoice Number: ({{Digits}})'
    regex: true
- name: dateIssued
  objectType: field
  fieldProperties:
    fieldType: macros
    expression: 'Date Issued: ({{SmartDate}})'
    regex: true
    dataType: date
    dateFormat: auto-mdy
- name: total
  objectType: field
  fieldProperties:
    fieldType: macros
    expression: 'Total: ({{Money}})'
    regex: true
    dataType: decimal
- name: currency
  objectType: field
  fieldProperties:
    fieldType: static
    expression: USD
    regex: true
- name: table1
  objectType: table
  tableProperties:
    start:
      expression: Description{{Spaces}}Hours
```

```
      regex: true
    end:
      expression: 'Total:'
      regex: true
    row:
      expression: '{{LineStart}}{{Spaces}}(?<description>{{SentenceWithSingleSpaces}})
      regex: true
    columns:
    - name: hours
      dataType: integer
    - name: unitPrice
      dataType: decimal
```

Module1.vb

```
Imports System.Collections.Specialized
Imports System.IO
Imports System.Net
Imports System.Text
Imports System.Threading
Imports Newtonsoft.Json.Linq

' Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://lo
' If it's not then please replace this with with your hosting url.
Module Module1

    ' The authentication key (API Key).
    ' Get your own by registering at https://app.pdf.co/documentation/api
    Const API_KEY As String = "*********************************"

    ' Source Input file
    Const SourceFile As String = ".\DigitalOcean-scanned.jpg"

    ' Destination TXT file name
    Const DestinationFile As String = ".\result.json"

    ' (!) Make asynchronous job
    Const Async As Boolean = True

    Sub Main()

        ' Template text. Use Document Parser SDK (https//bytescout.com/products/develo
        ' to create templates.
        ' Read template from file
        Dim templateText As String = File.ReadAllText("DigitalOcean.yml")


        ' Create standard .NET web client instance
        Dim webClient As WebClient = New WebClient()
```

```
        ' Set API Key
        webClient.Headers.Add("x-api-key", API_KEY)

        ' 1. RETRIEVE THE PRESIGNED URL TO UPLOAD THE FILE.
        ' * If you already have a direct file URL, skip to the step 3.

        ' Prepare URL for `Get Presigned URL` API call
        Dim query As String = Uri.EscapeUriString(String.Format(
            "https://localhost/file/upload/get-presigned-url?contenttype=application/o
            Path.GetFileName(SourceFile)))

        Try
            ' Execute request
            Dim response As String = webClient.DownloadString(query)

            ' Parse JSON response
            Dim json As JObject = JObject.Parse(response)

            If json("error").ToObject(Of Boolean) = False Then
                ' Get URL to use for the file upload
                Dim uploadUrl As String = json("presignedUrl").ToString()
                ' Get URL of uploaded file to use with later API calls
                Dim uploadedFileUrl As String = json("url").ToString()

                ' 2. UPLOAD THE FILE TO CLOUD.
                webClient.Headers.Add("content-type", "application/octet-stream")
                webClient.UploadFile(uploadUrl, "PUT", SourceFile) ' You can use Uploa

                ' 3. PARSE UPLOADED PDF DOCUMENT

                ' URL for `Document Parser` API call
                query = Uri.EscapeUriString(
                    String.Format("https://localhost/pdf/documentparser?url={0}&async=
                    )

                Dim requestBody As New NameValueCollection()
                requestBody.Add("template", templateText)

                ' Execute request
                Dim responseBytes As Byte() = webClient.UploadValues(query, "POST", re
                response = Encoding.UTF8.GetString(responseBytes)

                ' Parse JSON response
                json = JObject.Parse(response)

                If json("error").ToObject(Of Boolean) = False Then

                    ' Asynchronous job ID
                    Dim jobId As String = json("jobId").ToString()
                    ' URL of generated PDF file that will available after the job comp
                    Dim resultFileUrl As String = json("url").ToString()

                    ' Check the job status in a loop.
                    ' If you don't want to pause the main thread you can rework the co
                    ' to use a separate thread for the status checking and completion.
                    Do
                        Dim status As String = CheckJobStatus(jobId) ' Possible status

                        ' Display timestamp and status (for demo purposes)
```

```
                    Console.WriteLine(DateTime.Now.ToLongTimeString() + ": " + sta

                    If status = "success" Then

                        ' Download PDF file
                        webClient.DownloadFile(resultFileUrl, DestinationFile)

                        Console.WriteLine("Generated JSON file saved as ""{0}"" fi
                        Exit Do

                    ElseIf status = "working" Then

                        ' Pause for a few seconds
                        Thread.Sleep(3000)

                    Else

                        Console.WriteLine(status)
                        Exit Do

                    End If

                Loop

            Else
                Console.WriteLine(json("message").ToString())
            End If

        End If

    Catch ex As WebException
        Console.WriteLine(ex.ToString())
    End Try

    webClient.Dispose()

    Console.WriteLine()
    Console.WriteLine("Press any key...")
    Console.ReadKey()

End Sub


Function CheckJobStatus(jobId As String) As String

    Using webClient As WebClient = New WebClient()

        ' Set API Key
        webClient.Headers.Add("x-api-key", API_KEY)

        Dim url As String = "https://localhost/job/check?jobid=" + jobId

        Dim response As String = webClient.DownloadString(url)
        Dim json As JObject = JObject.Parse(response)

        Return Convert.ToString(json("status"))

    End Using

End Function
```

```
End Module
```

packages.config

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Newtonsoft.Json" version="10.0.3" targetFramework="net40" />
</packages>
```

VIDEO

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

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Cloud API Server Home Page
Explore ByteScout Cloud API Server Documentation
Explore Samples
Sign Up for ByteScout Cloud API Server Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com