

[www.bytescout.com](http://www.bytescout.com)

## document parser API in VB.NET and PDF.co Web API

PDF.co Web API is the Rest API that provides set of data extraction functions, tools for documents manipulation, splitting and merging of pdf files. Includes built-in OCR, images recognition, can generate and read barcodes from images, scans and pdf.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about PDF.co Web API](#)

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

## Source Code Files:

LifeAndAnnuityQuoteRequest.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}") = "LifeAndAnnuityQuoteRequest", "LifeAndAnnuityQuoteRequest.sln"
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|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {9B91124C-66C3-4BD9-B29E-168C1ABB15AC}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
EndGlobal
```

```

EndGlobalSection
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {4576C9BB-A42D-46A8-9198-7E2982E122FA}
EndGlobalSection
EndGlobal

```

## LifeAndAnnuityQuoteRequest.vbproj

```

<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''">AnyCPU</Platform>
    <ProjectGuid>{9B91124C-66C3-4BD9-B29E-168C1ABB15AC}</ProjectGuid>
    <OutputType>Exe</OutputType>
    <StartupObject>LifeAndAnnuityQuoteRequest.Module1</StartupObject>
    <RootNamespace>LifeAndAnnuityQuoteRequest</RootNamespace>
    <AssemblyName>LifeAndAnnuityQuoteRequest</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>LifeAndAnnuityQuoteRequest.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>LifeAndAnnuityQuoteRequest.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, PublicKeyToken=30ad4fe6b2a6aeed, p
    <HintPath>packages\Newtonsoft.Json.10.0.3\lib\net40\Newtonsoft.Json.dll</HintPath>
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
    <Reference Include="System.Xml" />
  </ItemGroup>

```

```

<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="SampleGroupDisabilityForm.yml">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</None>
<None Include="SampleGroupDisabilityForm.pdf">
  <CopyToOutputDirectory>Always</CopyToOutputDirectory>
</None>
<None Include="packages.config" />
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.VisualBasic.targets" />
</Project>

```

## Module1.vb

```

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

Module Module1

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

  ' Sample document containing life and annuity quote request form
  Const SourceFile As String = ".\SampleGroupDisabilityForm.pdf"

  ' PDF document password. Leave empty for unprotected documents.
  Const Password As String = ""

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

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

  Sub Main()

    ' Sample template
    Dim templateText As String = File.ReadAllText("SampleGroupDisabilityForm.yml")

    If ProcessAndGetJsonFields(templateText) Then

      ' Get json strings
      Dim jsonString As String = File.ReadAllText(DestinationFile)

```

```

' Parse to Json Fields
Dim oRet = ParseJsonFields(jsonString)

' Display some of data to console
Console.WriteLine({code}quot;Parsing Details:{Environment.NewLine}-----")

Console.WriteLine({code}quot;Contact Person: {oRet.ContactPerson}")
Console.WriteLine({code}quot;Business Name: {oRet.BusinessName}")
Console.WriteLine({code}quot;Business Address: {oRet.BusinessAddress}")
Console.WriteLine({code}quot;Business Type: {oRet.BusinessType}")
Console.WriteLine({code}quot;Phone: {oRet.Phone}")
Console.WriteLine({code}quot;Email: {oRet.Email}")

' Export list of census to csv format
Dim csvOutputFile As String = "result.csv"

Dim csvString = ConvertToCsv(oRet.lstCensusTable)
File.WriteAllText(csvOutputFile, csvString)

Console.WriteLine({code}quot;{Environment.NewLine}{csvOutputFile} generated successfully!")

```

End If

```

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

```

End Sub

```

''' <summary>

```

```

''' Process Input File and Get Json Fields

```

```

''' </summary>

```

```

Function ProcessAndGet.JsonFields(ByVal templateText As String) As Boolean

```

```

' 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://api.pdf.co/v1/file/upload/get-presigned-url?contenttype=application/octet-stream&name={0}",
    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 UploadData() instead if your file is byte a

```

```

' 3. PARSE UPLOADED PDF DOCUMENT

```

```

' URL for `Document Parser` API call
query = Uri.EscapeUriString(
    String.Format("https://api.pdf.co/v1/pdf/documentparser?url={0}&async={1}", uploadedFileUrl, Async)
)

```

```

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

```

```

' Execute request
Dim responseBytes As Byte() = webClient.UploadValues(query, "POST", requestBody)
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 completion
    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 code
    ' to use a separate thread for the status checking and completion.
    Do
        Dim status As String = CheckJobStatus(jobId) ' Possible statuses: "working", "failed", "aborted", "succe

        ' Display timestamp and status (for demo purposes)
        Console.WriteLine(DateTime.Now.ToLongTimeString() + ": " + status)

        If status = "success" Then

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

            Console.WriteLine("Generated JSON file saved as ""{0}"" file.", DestinationFile)

            Return True

        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()

Return False

End Function

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://api.pdf.co/v1/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

''' <summary>

```

```

''' Parse Json Fields
''' </summary>
Function ParseJsonFields(ByVal jsonInput As String) As FormVM

    Dim jsonObj As JObject = JObject.Parse(jsonInput)

    Dim oRet As New FormVM()

    oRet.ContactPerson = jsonObj.TraverseJObject("ContactPerson", "field")?.Value(Of String)("value")
    oRet.BusinessName = jsonObj.TraverseJObject("BusinessName", "field")?.Value(Of String)("value")
    oRet.BusinessAddress = jsonObj.TraverseJObject("BusinessAddress", "field")?.Value(Of String)("value")
    oRet.BusinessType = jsonObj.TraverseJObject("BusinessType", "field")?.Value(Of String)("value")
    oRet.Phone = jsonObj.TraverseJObject("Phone", "field")?.Value(Of String)("value")
    oRet.Email = jsonObj.TraverseJObject("Email", "field")?.Value(Of String)("value")

    Dim censusTable = jsonObj.TraverseJObject("CencusTable", "table")

    If Not censusTable Is Nothing Then

        Dim rows = censusTable("rows")

        If Not rows Is Nothing And rows.Count() > 0 Then

            For i As Int32 = 1 To rows.Count() - 1
                Dim oCensus As New CensusTableVm()

                ' Parse individual data
                oCensus.SrNo = rows.ElementAt(i).ElementAt(0).ElementAt(0).Value(Of String)("value")
                oCensus.DOB = rows.ElementAt(i).ElementAt(1).ElementAt(0).Value(Of String)("value")
                oCensus.Gender = rows.ElementAt(i).ElementAt(2).ElementAt(0).Value(Of String)("value")
                oCensus.Occupation = rows.ElementAt(i).ElementAt(3).ElementAt(0).Value(Of String)("value")
                oCensus.Salary = rows.ElementAt(i).ElementAt(4).ElementAt(0).Value(Of String)("value")
                oCensus.IsShortTermDisability = rows.ElementAt(i).ElementAt(5).ElementAt(0).Value(Of String)("value")
                oCensus.IsLongTernDisability = rows.ElementAt(i).ElementAt(6).ElementAt(0).Value(Of String)("value")
                oCensus.LifeInsuCoverAmt = rows.ElementAt(i).ElementAt(7).ElementAt(0).Value(Of String)("value")

                oRet.lstCensusTable.Add(oCensus)
            Next

        End If

    End If

    Return oRet

End Function

''' <summary>
''' Convert to Csv Format
''' </summary>
Function ConvertToCsv(ByVal lstCensusTable As List(Of CensusTableVm)) As String

    Dim oRet As New StringBuilder(String.Empty)

    ' Get Header Row
    oRet.AppendLine(getCsvTitleRow())

    ' Put Child Rows
    For Each item In lstCensusTable
        oRet.AppendLine(item.ToString())
    Next

    Return oRet.ToString()

End Function

Class FormVM

    Public Property ContactPerson As String
    Public Property BusinessName As String
    Public Property BusinessAddress As String
    Public Property BusinessType As String
    Public Property Phone As String
    Public Property Email As String

    Public Property lstCensusTable As List(Of CensusTableVm) = New List(Of CensusTableVm)()

End Class

```

## Class CensusTableVm

```
Public Property SrNo As String
Public Property DOB As String
Public Property Gender As String
Public Property Occupation As String
Public Property Salary As String
Public Property IsShortTermDisability As String
Public Property IsLongTernDisability As String
Public Property LifeInsuCoverAmt As String
```

```
Public Overrides Function ToString() As String
    Return {code}quot;{SrNo},{DOB},{Gender},{Occupation},{Salary},{IsShortTermDisability},{IsLongTernDisability}
End Function
```

End Class

```
Public Function getCsvTitleRow() As String
    Return "SrNo,DOB,Gender,Occupation,Salary,IsShortTermDisability,IsLongTermDisability,LifeInsuCoverAmt"
End Function
```

End Module

```
''' <summary>
```

```
''' Extension method container for JObject
```

```
''' </summary>
```

```
Module JObjectExtensionMethods
```

```
<Extension(>
```

```
Public Function TraverseJObject(ByVal jsonObj As JObject, ByVal propertyName As String, ByVal objectType As S
    Return jsonObj("objects").Where(Function(x) x.Value(Of String)("name").Equals(propertyName) AndAlso x.Value
```

```
End Function
```

End Module

## SampleGroupDisabilityForm.yml

```
templateVersion: 3
templatePriority: 0
sourceId: Untitled document kind
detectionRules:
  keywords: []
fields:
  CensusTable:
    type: rectangle
    dataType: table
    rectangle:
      - 27
      - 324.75
      - 554.25
      - 358.5
    pageIndex: 0
    rowMergingRule: byBorders
  ContactPerson:
    type: rectangle
    rectangle:
      - 178.5
      - 111
      - 76.5
      - 15
    pageIndex: 0
  BusinessName:
    type: rectangle
    rectangle:
      - 177.75
```

```
- 126
- 105
- 17.25
pageIndex: 0
BusinessAddress:
  type: rectangle
  rectangle:
    - 183
    - 144.75
    - 187.5
    - 24
pageIndex: 0
BusinessType:
  type: rectangle
  rectangle:
    - 183.75
    - 206.25
    - 77.25
    - 19.5
pageIndex: 0
Phone:
  type: rectangle
  rectangle:
    - 181.5
    - 236.25
    - 86.25
    - 15
pageIndex: 0
Email:
  type: rectangle
  rectangle:
    - 182.25
    - 250.5
    - 112.5
    - 17.25
pageIndex: 0
```

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 PDF.co Web API Home Page](#)  
[Explore PDF.co Web API Documentation](#)  
[Explore Samples](#)  
[Sign Up for PDF.co Web API 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](http://www.bytescout.com)