

How to parse from uploaded file for document parser API in Powershell and PDF.co Web API

Step-by-step tutorial:How to parse from uploaded file to have document parser API in Powershell

Quick guide:Learn how to parse from uploaded file in Powershell. PDF.co Web API was designed to assist document parser API in Powershell. 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.

If you want to speed up the application's code writing then Powershell code samples for Powershell developers help to implement using PDF.co Web API. For implementation of this functionality, please copy and paste the code below into your app using code editor. Then compile and run your app. Further improvement of the code will make it more robust.

PDF.co Web API - free trial version is available on our website. Also, there are other code samples to help you with your Powershell application included into trial version.

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:

MultiPageTable-template1.yml

```
---
# Template that demonstrates parsing of multi-page table using only
# regular expressions for the table start, end, and rows.
# If regular expression cannot be written for every table row (for example,
# if the table contains empty cells), try the second method demonstrated
# in 'MultiPageTable-template2.yml' template.
templateVersion: 2
templatePriority: 0
sourceId: Multipage Table Test
detectionRules:
  keywords:
    - Sample document with multi-page table
  fields:
    total:
      expression: TOTAL {{DECIMAL}}
  tables:
    - name: table1
      start:
        # regular expression to find the table start in document
        expression: Item\s+Description\s+Price\s+Qty\s+Extended Price
      end:
        # regular expression to find the table end in document
        expression: TOTAL\s+\d+\.\d\d
      row:
        # regular expression to find table rows
        expression: '^s*(?<itemNo>\d+)\s+(?<description>.+)\s+(?<price>\d+\.\d\d)\s+(?<qty>\d+)\s+(?<extPrice>\d+\.\d\d)'
      columns:
        - name: itemNo
          type: integer
        - name: description
          type: string
        - name: price
          type: decimal
        - name: qty
          type: integer
        - name: extPrice
          type: decimal
  multipage: true
```

ParseFromUploadedFile.ps1

```
# The authentication key (API Key).
# Get your own by registering at https://app.pdf.co/documentation/api
$API_KEY = "*****"

# Source PDF file
$SourceFile = ".\MultiPageTable.pdf"
```

```

# Destination JSON file name
$DestinationFile = ".\result.json"

# 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
$query = "https://api.pdf.co/v1/file/upload/get-presigned-url?contenttype=application/
[System.IO.Path]::GetFileName($SourceFile)
$query = [System.Uri]::EscapeUriString($query)

try {
    # Execute request
    $jsonResponse = Invoke-RestMethod -Method Get -Headers @{ "x-api-key" = $API_KEY }

    if ($jsonResponse.error -eq $false) {
        # Get URL to use for the file upload
        $uploadUrl = $jsonResponse.presignedUrl
        # Get URL of uploaded file to use with later API calls
        $uploadedFileUrl = $jsonResponse.url

        # 2. UPLOAD THE FILE TO CLOUD.

        $r = Invoke-WebRequest -Method Put -Headers @{ "x-api-key" = $API_KEY; "content

        if ($r.StatusCode -eq 200) {

            # 3. Parse PDF document by template

            # Template text. Use Document Parser SDK (https://bytescout.com/products/d
            # to create templates.
            # Read template from file:
            $templateContent = [IO.File]::ReadAllText(".\MultiPageTable-template1.yml")

            # Prepare URL for `Document Parser` API call
            $query = "https://api.pdf.co/v1/pdf/documentparser"

            # Content
            $Body = @{
                "url" = $uploadedFileUrl;
                "template" = $templateContent;
            }

            # Execute request
            $jsonResponse = Invoke-RestMethod -Method 'Post' -Headers @{ "x-api-key" =

            if ($jsonResponse.error -eq $false) {
                # Get URL of generated HTML file
                $resultFileUrl = $jsonResponse.url;

                # Download output file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile $Destin

                Write-Host "Generated output file saved as `"$($DestinationFile)`" file
            }
            else {
                # Display service reported error
                Write-Host $jsonResponse.message
            }
        }
    }
}

```

```
    }
    else {
        # Display request error status
        Write-Host $r.StatusCode + " " + $r.StatusDescription
    }
}
else {
    # Display service reported error
    Write-Host $jsonResponse.message
}
}
catch {
    # Display request error
    Write-Host $_.Exception
}
```

run.bat

```
@echo off

powershell -NoProfile -ExecutionPolicy Bypass -Command "& .\ParseFromUploadedFile.ps1"
echo Script finished with errorlevel=%errorlevel%

pause
```

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