www.bytescout.com

How to PDF text search API in PowerShell and ByteScout Cloud API Server

Learn to code in PowerShell to PDF text search API with this step-by-step tutorial

The coding instructions are formulated to help you to try-out the features without the requirement to write your own code. ByteScout Cloud API Server is API server that is ready to use and can be installed and deployed in less than 30 minutes on your own Windows server or server in a cloud. It can save data and files on your local server-based file storage or in Amazon AWS S3 storage. Data is processed solely on the API server and is powered by ByteScout engine, no cloud services or Internet connection is required for data processing. and you can use it to PDF text search API with PowerShell.

Want to save time? You will save a lot of time on writing and testing code as you may just take the PowerShell code from ByteScout Cloud API Server for PDF text search API below and use it in your application. Simply copy and paste in your PowerShell project or application you and then run your app! Use of ByteScout Cloud API Server in PowerShell is also described in the documentation included along with the product.

ByteScout Cloud API Server free trial version is available on our website. PowerShell and other programming languages are supported.

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:

PDFTextSearchFromUploadedFile.ps1

```
# 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.

# Source file name
$SourceFile = ".\sample.pdf"

# Comma-separated list of page indices (or ranges) to process. Leave empty for all pag
$Pages = ""

# PDF document password. Leave empty for unprotected documents.
$Password = ""

# Search string.
$SearchString = '\d{1,}\.\d\d' #Regular expression to find numbers like '100.00'

# Enable regular expressions (Regex)
$RegexSearch = 'True'

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

try {
    # Execute request
    $jsonResponse = Invoke-RestMethod -Method Get  -Uri $query

    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 @{ "content-type" = "application/o

        if ($r.StatusCode -eq 200) {

            # 3. TEXT SEARCH FROM UPLOADED FILE

            # Prepare URL for PDF text search API call.
            $query = "https://localhost/pdf/find?password=$($Password)&pages=$($Pages)&
            $query = [System.Uri]::EscapeUriString($query)

            # Execute request
            $jsonResponse = Invoke-RestMethod -Method Get  -Uri $query

            if ($jsonResponse.error -eq $false) {
                # Display found barcodes in console
```

```
                foreach ($item in $jsonResponse.body)
                {
                    Write-Host "Found text $($item.text) at coordinates $($item.left),
                }
            }
            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 "& .\PDFTextSearchFromUploadedF
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 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

visit www.PDF.co