www.bytescout.com

How to convert PDF to text from URL asynchronously for PDF to text API in PowerShell using ByteScout Cloud API Server

Learn in simple ways: How to convert PDF to text from URL asynchronously for PDF to text API in PowerShell

Check these thousands of pre-made source code samples for simple implementation in your own programming projects. ByteScout Cloud API Server was designed to assist PDF to text API in PowerShell. ByteScout Cloud API Server is 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!.

PowerShell code snippet like this for ByteScout Cloud API Server works best when you need to quickly implement PDF to text API in your PowerShell application. This PowerShell sample code can be used by copying and pasting into your project. Once done,just compile your project and click Run. You can use these PowerShell sample examples in one or many applications.

ByteScout Cloud API Server - 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 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:

ConvertPdfToTextFromUrlAsynchronously.ps1

```
# Cloud API asynchronous "PDF To Text" job example.
# Allows to avoid timeout errors when processing huge or scanned PDF documents.

# 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.

# Direct URL of source PDF file.
$SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pd
# 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 = ""
# Destination TXT file name
$DestinationFile = ".\result.txt"
# (!) Make asynchronous job
$Async = $true

# Prepare URL for `PDF To Text` API call
$query = "https://localhost/pdf/convert/to/text?name={0}&password={1}&pages={2}&url={3
    $(Split-Path $DestinationFile -Leaf), $Password, $Pages, $SourceFileUrl, $Async
$query = [System.Uri]::EscapeUriString($query)

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

    if ($jsonResponse.error -eq $false) {
        # Asynchronous job ID
        $jobId = $jsonResponse.jobId
        # URL of generated TXT file that will available after the job completion
        $resultFileUrl = $jsonResponse.url

        # Check the job status in a loop.
        do {
            $statusCheckUrl = "https://localhost/job/check?jobid=" + $jobId
            $jsonStatus = Invoke-RestMethod -Method Get  -Uri $statusCheckUrl

            # Display timestamp and status (for demo purposes)
            Write-Host "$(Get-date): $($jsonStatus.status)"

            if ($jsonStatus.status -eq "success") {
                # Download TXT file
                Invoke-WebRequest  -OutFile $DestinationFile -Uri $resultFileUrl
                Write-Host "Generated TXT file saved as `"$($DestinationFile)`" file."
                break
            }
            elseif ($jsonStatus.status -eq "working") {
                # Pause for a few seconds
                Start-Sleep -Seconds 3
            }
            else {
                Write-Host $jsonStatus.status
                break
```

```
            }
        }
        while ($true)
    }
    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 "& .\ConvertPdfToTextFromUrlAsy
echo Script finished with errorlevel=%errorlevel%

pause
```

VIDEO

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

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

visit www.ByteScout.com

visit www.PDF.co