How to convert PDF to TIFF from URL asynchronously for PDF to image API in PowerShell with PDF.co Web API

See how to convert PDF to TIFF from URL asynchronously to have PDF to image API in PowerShell

The sample source codes on this page will demonstrate you how to make PDF to image API in PowerShell. PDF.co Web API helps with PDF to image 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.

You will save a lot of time on writing and testing code as you may just take the code below and use it in your application. Open your PowerShell project and simply copy & paste the code and then run your app! Further enhancement of the code will make it more vigorous.

ByteScout free trial version is available for FREE download from our website. Programming tutorials along with source code samples are included.

PowerShell - ConvertPdfToTiffFromUrlAsynchronously.ps1

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

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

# Direct URL of source PDF file.
$SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-to-image/sample.pdf"
# Comma-separated list of page indices (or ranges) to process. Leave empty for all pages. Example: '0,2-5,7-'.
$Pages = ""
# PDF document password. Leave empty for unprotected documents.
$Password = ""
# Destination TIFF file name
$DestinationFile = ".\result.tif"
# (!) Make asynchronous job
$Async = $true


# Prepare URL for `PDF To TIFF` API call
$query = "https://api.pdf.co/v1/pdf/convert/to/tiff?name={0}&password={1}&pages={2}&url={3}&async={4}" -f `
```

```
    $(Split-Path $DestinationFile -Leaf), $Password, $Pages, $SourceFileUrl, $Async
$query = [System.Uri]::EscapeUriString($query)

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

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

        # Check the job status in a loop.
        do {
            $statusCheckUrl = "https://api.pdf.co/v1/job/check?jobid=" + $jobId
            $jsonStatus = Invoke-RestMethod -Method Get -Headers @{ "x-api-key" =
$API_KEY } -Uri $statusCheckUrl

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

            if ($jsonStatus.status -eq "success") {
                # Download TIFF file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile
$DestinationFile -Uri $resultFileUrl
                Write-Host "Generated TIFF file saved as `"$($DestinationFile)`""
file."
                break
            }
            elseif ($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
}
```

```
@echo off

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

pause
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about PDF.co Web API]

[Explore documentation]

[Visit www.ByteScout.com]

or

[Get Your Free API Key for www.PDF.co Web API]