www.bytescout.com

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

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

These source code samples are listed and grouped by their programming language and functions they use. PDF.co Web API was made to help with PDF to HTML 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.

This rich sample source code in PowerShell for PDF.co Web API includes the number of functions and options you should do calling the API to implement PDF to HTML API. This PowerShell sample code should be copied and pasted into your project. After doing this just compile your project and click Run. Enjoy writing a code with ready-to-use sample PowerShell codes to add PDF to HTML API functions using PDF.co Web API in PowerShell.

PDF.co Web API - free trial version is on available 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

Source Code Files:

ConvertPdfToHtmlFromUrlAsynchronously.ps1

```
# Cloud API asynchronous "PDF To HTML" 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
# Comma-separated list of page indices (or ranges) to process. Leave empty for all page
$Pages = ""
# PDF document password. Leave empty for unprotected documents.
$Password = ""
# Destination HTML file name
$DestinationFile = ".\result.html"
# Set to $true to get simplified HTML without CSS. Default is the rich HTML keeping the
$PlainHtml = $false
# Set to $true if your document has the column layout like a newspaper.
$ColumnLayout = $false
# (!) Make asynchronous job
$Async = $true


# Prepare URL for `PDF To HTML` API call
$query = "https://api.pdf.co/v1/pdf/convert/to/html?name={0}&password={1}&pages={2}&sim
    $(Split-Path $DestinationFile -Leaf), $Password, $Pages, $PlainHtml, $ColumnLayout
$query = [System.Uri]::EscapeUriString($query)

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

    if ($jsonResponse.error -eq $false) {
        # Asynchronous job ID
        $jobId = $jsonResponse.jobId
        # URL of generated HTML 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_

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

            if ($jsonStatus.status -eq "success") {
                # Download HTML file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile $Destin
                Write-Host "Generated HTML file saved as `"$($DestinationFile)`" file."
                break
            }
            elseif ($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 "& .\ConvertPdfToHtmlFromUrlAsyn
echo Script finished with errorlevel=%errorlevel%

pause
```

VIDEO

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

ON-PREMISE OFFLINE SDK