How to split PDF from URL asynchronously for PDF splitting API in PowerShell and PDF.co Web API

How to split PDF from URL asynchronously for PDF splitting API in PowerShell: How To Tutorial

Source code documentation samples provide quick and easy way to add a required functionality into your application. PDF.co Web API was made to help with PDF splitting 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.

The SDK samples like this one below explain how to quickly make your application do PDF splitting API in PowerShell with the help of PDF.co Web API. Sample code in PowerShell is all you need. Copy-paste it to your the code editor, then add a reference to PDF.co Web API and you are ready to try it! Writing PowerShell application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

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

PowerShell - SplitPdfFromUrlAsynchronously.ps1

```powershell
# Cloud API asynchronous "Split PDF" 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 = "***********************************"

# Source PDF file to split
$SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-split/sample.pdf"
# Comma-separated list of page numbers (or ranges) to process. Example: '1,3-5,7-'.
$Pages = "1-2,3-"
# (!) Make asynchronous job
$Async = $true


# Prepare URL for `Split PDF` API call
$query = "https://api.pdf.co/v1/pdf/split?pages=$($Pages)&url=$($SourceFileUrl)&async=$($Async)"
$query = [System.Uri]::EscapeUriString($query)

try {
```

```powershell
    # 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 JSON file available after the job completion; it will
contain URLs of result PDF files.
        $resultJsonFileUrl = $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 JSON file with URLs of result PDF files
                $jsonPdfUrls = Invoke-RestMethod -Method Get -Headers @{ "x-api-key"
= $API_KEY } -Uri $resultJsonFileUrl

                # Download generated PDF files
                $part = 1;
                foreach ($url in $jsonPdfUrls) {
                    $localFileName = ".\part$($part).pdf"

                    # Download PDF file
                    Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile
$localFileName -Uri $url

                    Write-Host "Downloaded `"$($localFileName)`""
                    $part++
                }
                break
            }
            elseif ($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
}
```

PowerShell - run.bat

```
@echo off

powershell -NoProfile -ExecutionPolicy Bypass -Command "&
.\SplitPdfFromUrlAsynchronously.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