How to merge PDF documents from urls asynchronously for PDF merging API in PowerShell and PDF.co Web API

How to merge PDF documents from urls asynchronously in PowerShell with easy ByteScout code samples to make PDF merging API. Step-by-step tutorial

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 merging API in PowerShell. PDF.co Web API is the flexible Web API that includes full set of functions from e-signature requests to data extraction, OCR, images recognition, pdf splitting and pdf splitting. Can also generate barcodes and read barcodes from images, scans and pdf.

PowerShell code samples for PowerShell developers help to speed up the application's code writing when using PDF.co Web API. 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.

Trial version of ByteScout is available for free download from our website. This and other source code samples for PowerShell and other programming languages are available.

PowerShell - MergePdfDocumentsFromUrlsAsynchronously.ps1

```
# Cloud API asynchronous "Merge 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 = "**********************************"

# Direct URLs of PDF documents to merge
$SourceFiles = @(
    "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-merge/sample1.pdf",
    "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-merge/sample2.pdf"
)
# Destination PDF file name
$DestinationFile = ".\result.pdf"
# (!) Make asynchronous job
$Async = $true


# Prepare URL for `Merge PDF` API call
$query = "https://api.pdf.co/v1/pdf/merge?name={0}&url={1}&async={2}" -f `
    $(Split-Path $DestinationFile -Leaf), $($SourceFiles -join ","), $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 PDF 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 PDF file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile
$DestinationFile -Uri $resultFileUrl
                Write-Host "Generated PDF file saved as `"$($DestinationFile)`"
file."
                break
            }
            elseif ($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 "&
.\MergePdfDocumentsFromUrlsAsynchronously.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]