

How to convert PDF to XLSX from URL asynchronously for PDF to excel API in PowerShell and PDF.co Web API

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

These source code samples are listed and grouped by their programming language and functions they use. PDF to excel API in PowerShell can be implemented with PDF.co Web API. PDF.co Web API is the Web API with a set of tools for documents manipulation, data conversion, data extraction, splitting and merging of documents. Includes image recognition, built-in OCR, barcode generation and barcode decoders to decode bar codes from scans, pictures and pdf.

The SDK samples like this one below explain how to quickly make your application do PDF to excel API in PowerShell with the help of PDF.co Web API. This PowerShell sample code should be copied and pasted into your project. After doing this just compile your project and click Run. You can use these PowerShell sample examples in one or many applications.

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.

PowerShell - ConvertPdfToXlsxFromUrlAsynchronously.ps1

```
# Cloud API asynchronous "PDF To XLSX" 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-excel/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 XLSX file name
$DestinationFile = ".\result.xlsx"
# (!) Make asynchronous job
$Async = $true

# Prepare URL for `PDF To XLSX` API call
$query = "https://api.pdf.co/v1/pdf/convert/to/xlsx?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 XLSX 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 XLSX file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile
$DestinationFile -Uri $resultFileUrl
                Write-Host "Generated XLSX file saved as `"$($DestinationFile)`"
file."
                break
            }
            elseif ($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 "&  
. \ConvertPdfToXlsxFromUrlAsynchronously.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](#)