

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

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

Every ByteScout tool contains example PowerShell source codes that you can find here or in the folder with installed ByteScout product. PDF to XML API in PowerShell can be implemented with PDF.co Web API. 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 snippet like this for PDF.co Web API works best when you need to quickly implement PDF to XML API in your PowerShell application. 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.

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.

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](http://www.ByteScout.com)

Source Code Files:

ConvertPdfToXmlFromUrlAsynchronously.ps1

```
# Cloud API asynchronous "PDF To XML" 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 pages.
$Pages = ""
# PDF document password. Leave empty for unprotected documents.
$Password = ""
# Destination XML file name
$DestinationFile = ".\result.xml"
# (!) Make asynchronous job
$Async = $true

# Prepare URL for `PDF To XML` API call
$query = "https://api.pdf.co/v1/pdf/convert/to/xml?name={0}&password={1}&pages={2}&url={3}"
$(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 }

    if ($jsonResponse.error -eq $false) {
        # Asynchronous job ID
        $jobId = $jsonResponse.jobId
        # URL of generated XML file that will be 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 }

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

            if ($jsonStatus.status -eq "success") {
                # Download XML file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile $DestinationFile
                Write-Host "Generated XML file saved as `"$($DestinationFile)`" file."
                break
            }
            elseif ($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 "& .\ConvertPdfToXmlFromUrlAsync.ps1"
echo Script finished with errorlevel=%errorlevel%

pause
```

VIDEO

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

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit PDF.co Web API Home Page](#)
[Explore PDF.co Web API Documentation](#)
[Explore Samples](#)
[Sign Up for PDF.co Web API Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)

[Explore Web API Docs](#)

[Explore Web API Samples](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com