

PDF to XML API in PowerShell with PDF.co Web API

Make PDF to XML API in PowerShell

:

Tutorial on how to do PDF to XML API in PowerShell

Writing of the code to PDF to XML API in PowerShell can be done by developers of any level using PDF.co Web API. PDF.co Web API was made to help with PDF to XML 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.

You will save a lot of time on writing and testing code as you may just take the code below and use it in your application. PowerShell sample code is all you need: copy and paste the code to your PowerShell application's code editor, add a reference to PDF.co Web API (if you haven't added yet) and you are ready to go! Test PowerShell sample code examples whether they respond your needs and requirements for the project.

On our website you may get trial version of PDF.co Web API for free. Source code samples are included to help you with your PowerShell application.

PowerShell - 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-to-xml/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 XML file name
$DestinationFile = ".\result.xml"
# (!) Make asynchronous job
$Async = $true
```

```

# Sample profile that sets advanced conversion options
# Advanced options are properties of XMLExtractor class from ByteScout XML Extractor
SDK used in the back-end:
# https://cdn.bytescout.com/help/BytescoutPDFExtractorSDK/html/6f2b5e9c-ba15-f9fe-
192b-c3e31ec4a0ee.htm
$Profiles = '{ "profiles": [ { "profile1": { "TrimSpaces": "False",
"PreserveFormattingOnTextExtraction": "True", "Unwrap": "True",
"ShrinkMultipleSpaces": "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}&async={4}&profiles={5}" -f `
    $(Split-Path $DestinationFile -Leaf), $Password, $Pages, $SourceFileUrl, $Async,
$Profiles
$query = [System.Uri]::EscapeUriString($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 XML 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 XML file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile
$DestinationFile -Uri $resultFileUrl
                Write-Host "Generated XML file saved as `"$($DestinationFile)`"
file."
                break
            }
            elseif ($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  
}
```

PowerShell - run.bat

```
@echo off  
  
powershell -NoProfile -ExecutionPolicy Bypass -Command "&  
.\ConvertPdfToXmlFromUrlAsynchronously.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](#)