

## How to convert PDF to XLSX from URL asynchronously for PDF to excel API in PowerShell and ByteScout Cloud API Server

Follow this simple tutorial to learn convert PDF to XLSX from URL asynchronously to have PDF to excel API in PowerShell

Quick guide: Learn how to convert PDF to XLSX from URL asynchronously in PowerShell. PDF to excel API in PowerShell can be applied with ByteScout Cloud API Server. ByteScout Cloud API Server is API server that is ready to use and can be installed and deployed in less than 30 minutes on your own Windows server or server in a cloud. It can save data and files on your local server-based file storage or in Amazon AWS S3 storage. Data is processed solely on the API server and is powered by ByteScout engine, no cloud services or Internet connection is required for data processing..

PowerShell code snippet like this for ByteScout Cloud API Server works best when you need to quickly implement PDF to excel API in your PowerShell application. Follow the tutorial and copy - paste code for PowerShell into your project's code editor. This basic programming language sample code for PowerShell will do the whole work for you in implementing PDF to excel API in your app.

ByteScout Cloud API Server - free trial version is available on our website. Also, there are other code samples to help you with your PowerShell application included into trial version.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Cloud API Server](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Cloud API Server](#)

[Get Free API key for Web API](#)

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

Source Code Files:

## ConvertPdfToXlsxFromUrlAsynchronously.ps1

```
# Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://lo
# If it's not then please replace this with with your hosting url.

# Cloud API asynchronous "PDF To XLSX" job example.
# Allows to avoid timeout errors when processing huge or scanned PDF documents.

# 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 page
$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://localhost/pdf/convert/to/xlsx?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 -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://localhost/job/check?jobid=" + $jobId
            $jsonStatus = Invoke-RestMethod -Method Get -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 -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
}
```

run.bat

```
@echo off

powershell -NoProfile -ExecutionPolicy Bypass -Command "& .\ConvertPdfToXlsxFromUrlAsync.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 ByteScout Cloud API Server Home Page](#)  
[Explore ByteScout Cloud API Server Documentation](#)  
[Explore Samples](#)  
[Sign Up for ByteScout Cloud API Server 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](http://www.bytescout.com)