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

Learn in simple ways: How to convert PDF to XLSX from URL for PDF to excel API in PowerShell

Every ByteScout tool includes sample PowerShell source codes that you can find here or in the folder with installed ByteScout product. ByteScout Cloud API Server was designed to assist PDF to excel API in PowerShell. 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. This sample code in PowerShell is all you need. Just copy-paste it to the code editor, then add a reference to ByteScout Cloud API Server and you are ready to try it! You can use these PowerShell sample examples in one or many applications.

Free! Free! Free! ByteScout free trial version is available for FREE download from our website. Programming tutorials along with source code samples are assembled.

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

Source Code Files:

ConvertPdfToXlsxFromUrl.ps1

```powershell
# 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.

# Direct URL of source PDF file.
$SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pd
# Comma-separated list of page indices (or ranges) to process. Leave empty for all pag
$Pages = ""
# PDF document password. Leave empty for unprotected documents.
$Password = ""
# Destination XLSX file name
$DestinationFile = ".\result.xlsx"


# 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
$query = [System.Uri]::EscapeUriString($query)

try {
    # Execute request
    $jsonResponse = Invoke-RestMethod -Method Get  -Uri $query

    if ($jsonResponse.error -eq $false) {
        # Get URL of generated XLSX file
        $resultFileUrl = $jsonResponse.url;

        # Download XLSX file
        Invoke-WebRequest  -OutFile $DestinationFile -Uri $resultFileUrl

        Write-Host "Generated XLSX file saved as `"$($DestinationFile)`" file."
    }
    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 "& .\ConvertPdfToXlsxFromUrl.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

visit www.PDF.co