www.bytescout.com

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

Step By Step Instructions on how to convert PDF to XLS from URL for PDF to excel API in PowerShell

This page displays the step-by-step instructions and algorithm of how to convert PDF to XLS from URL and how to apply it in your application. ByteScout Cloud API Server was designed to assist PDF to excel API in PowerShell. ByteScout Cloud API Server is the ready to use Web API Server that can be deployed in less than 30 minutes into your own in-house server or into private cloud server. Can store data on in-house local server based storage or in Amazon AWS S3 bucket. Processing data solely on the server using buil-in ByteScout powered engine, no cloud services are used to process your data!.

Want to learn quickly? These fast application programming interfaces of ByteScout Cloud API Server for PowerShell plus the instruction and the code below will help to learn how to convert PDF to XLS from URL. 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! Further improvement of the code will make it more robust.

Our website provides free trial version of ByteScout Cloud API Server that gives source code samples to assist with your PowerShell project.

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:

ConvertPdfToXlsFromUrl.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.

# 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 XLS file name
$DestinationFile = ".\result.xls"


# Prepare URL for `PDF To XLS` API call
$query = "https://localhost/pdf/convert/to/xls?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 XLS file
        $resultFileUrl = $jsonResponse.url;

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

        Write-Host "Generated XLS 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 "& .\ConvertPdfToXlsFromUrl.ps1
echo Script finished with errorlevel=%errorlevel%

pause
```

VIDEO

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