How to convert DOC to PDF from URL asynchronously for DOC to PDF API in PowerShell with PDF.co Web API

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

The coding tutorials are designed to help you test the features without need to write your own code. DOC to PDF 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 samples for PowerShell developers help to speed up the application's code writing when using PDF.co Web API. Open your PowerShell project and simply copy & paste the code and then run your app! Use of PDF.co Web API in PowerShell is also explained in the documentation included along with the product.

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

PowerShell - ConvertDocToPdfFromUrlAsynchronously.ps1

```
# Cloud API asynchronous "DOC To PDF" 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 DOC or DOCX file.
$SourceFileURL = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-
api/doc-to-pdf/sample.docx"
# Destination PDF file name
$DestinationFile = ".\result.pdf"
# (!) Make asynchronous job
$Async = $true


# Prepare URL for `DOC To PDF` API call
$query = "https://api.pdf.co/v1/pdf/convert/from/doc?name=$(Split-Path
$DestinationFile -Leaf)&url=$($SourceFileURL)&async=$($Async)"
$query = [System.Uri]::EscapeUriString($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 PDF 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 PDF file
                Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile
$DestinationFile -Uri $resultFileUrl
                Write-Host "Generated PDF file saved as `"$($DestinationFile)`"
file."
                break
            }
            elseif ($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 "& .\ConvertDocToPdfFromUrlAsynchronously.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