How to optimize PDF from URL asynchronously for PDF optimization API in PowerShell and ByteScout Cloud API Server

Learn in simple ways: How to optimize PDF from URL asynchronously for PDF optimization API in PowerShell

Quick guide:Learn how to optimize PDF from URL asynchronously in PowerShell. ByteScout Cloud API Server helps with PDF optimization API in PowerShell. ByteScout Cloud API Server is the ready to deploy Web API Server that can be deployed in less than thirty minutes into your own in-house Windows server (no Internet connnection is required to process data!) 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 built-in ByteScout powered engine, no cloud services are used to process your data!.

If you want to speed up the application's code writing then PowerShell code samples for PowerShell developers help to implement using ByteScout Cloud API Server. 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! Enjoy writing a code with ready-to-use sample PowerShell codes to add PDF optimization API functions using ByteScout Cloud API Server in PowerShell.

Trial version of ByteScout is available for free download from our website. This and other source code samples for PowerShell and other programming languages are available.

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:

OptimizePdfFromUrlAsynchronously.ps1

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

# Cloud API asynchronous "Optimize PDF" 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/pd
# PDF document password. Leave empty for unprotected documents.
$Password = ""
# Destination PDF file name
$DestinationFile = ".\result.pdf"
# (!) Make asynchronous job
$Async = $true

# Prepare URL for `Optimize PDF` API call
$query = "https://localhost/pdf/optimize?name={0}&password={1}&url={2}&async={3}" -f `
    $(Split-Path $DestinationFile -Leaf), $Password, $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 PDF 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 PDF file
                Invoke-WebRequest  -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
}
```

run.bat

```
@echo off

powershell -NoProfile -ExecutionPolicy Bypass -Command "& .\OptimizePdfFromUrlAsynchron
echo Script finished with errorlevel=%errorlevel%

pause
```

VIDEO

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)