

PDF password and security in PowerShell with ByteScout Cloud API Server

What is ByteScout Cloud API Server? It 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..

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:

ProtectPDFDocument.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.

# The authentication key (API Key).
# Get your own by registering at https://app.pdf.co/documentation/api
$API_KEY = "*****"

# Direct URL of source PDF file.
$SourceFileURL = "https://bytescout-com.s3-us-west-2.amazonaws.com/files/demo-files/cl
# Destination PDF file name
$DestinationFile = ".\protected.pdf"

# Passwords to protect PDF document
# The owner password will be required for document modification.
# The user password only allows to view and print the document.
$OwnerPassword = "123456"
$UserPassword = "654321"

# Encryption algorithm.
# Valid values: "RC4_40bit", "RC4_128bit", "AES_128bit", "AES_256bit".
$EncryptionAlgorithm = "AES_128bit"

# Allow or prohibit content extraction for accessibility needs.
$AllowAccessibilitySupport = $True

# Allow or prohibit assembling the document.
$AllowAssemblyDocument = $True

# Allow or prohibit printing PDF document.
$AllowPrintDocument = $True

# Allow or prohibit filling of interactive form fields (including signature fields) in
$AllowFillForms = $True

# Allow or prohibit modification of PDF document.
$AllowModifyDocument = $True

# Allow or prohibit copying content from PDF document.
$AllowContentExtraction = $True

# Allow or prohibit interacting with text annotations and forms in PDF document.
$AllowModifyAnnotations = $True

# Allowed printing quality.
# Valid values: "HighResolution", "LowResolution"
$PrintQuality = "HighResolution"

# Runs processing asynchronously.
# Returns Use JobId that you may use with /job/check to check state of the processing
$async = $False

# Prepare URL for `PDF Security` API call
$query = "https://localhost/pdf/security/add"

# Prepare request body (will be auto-converted to JSON by Invoke-RestMethod)
# See documentation: https://apidocs.pdf.co
$body = @{"name" = $(Split-Path $DestinationFile -Leaf)
```

```

"url" = $SourceFileURL
"ownerPassword" = $OwnerPassword
"userPassword" = $UserPassword
"encryptionAlgorithm" = $EncryptionAlgorithm
"allowAccessibilitySupport" = $AllowAccessibilitySupport
"allowAssemblyDocument" = $AllowAssemblyDocument
"allowPrintDocument" = $AllowPrintDocument
"allowFillForms" = $AllowFillForms
"allowModifyDocument" = $AllowModifyDocument
"allowContentExtraction" = $AllowContentExtraction
"allowModifyAnnotations" = $AllowModifyAnnotations
"printQuality" = $PrintQuality
"async" = $async
} | ConvertTo-Json

try {
    # Execute request
    $response = Invoke-WebRequest -Method Post -Headers @{ "x-api-key" = $API_KEY; "Co

    $jsonResponse = $response.Content | ConvertFrom-Json

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

        # Download PDF file
        Invoke-WebRequest -Headers @{ "x-api-key" = $API_KEY } -OutFile $DestinationFi

        Write-Host "Generated PDF 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 "& .\ProtectPDFDocument.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