

How to PDF make searchable API in PowerShell and PDF.co Web API

Write code in PowerShell to PDF make searchable API with this step-by-step tutorial

We made thousands of pre-made source code pieces for easy implementation in your own programming projects. 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. It can PDF make searchable API in PowerShell.

PowerShell code samples for PowerShell developers help to speed up coding of your application when using PDF.co Web API. This PowerShell sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Implementing PowerShell application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Download free trial version of PDF.co Web API from our website with this and other source code samples for PowerShell.

PowerShell - MakeSearchablePdfFromUploadedFile.ps1

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

# Source PDF file
$SourceFile = ".\sample.pdf"
# Comma-separated list of page indices (or ranges) to process. Leave empty for all
pages. Example: '0,2-5,7-'.
$Pages = ""
# PDF document password. Leave empty for unprotected documents.
$Password = ""
# OCR language. "eng", "fra", "deu", "spa" supported currently. Let us know if you
need more.
$Language = "eng"
# Destination PDF file name
$DestinationFile = ".\result.pdf"

# 1. RETRIEVE THE PRESIGNED URL TO UPLOAD THE FILE.
# * If you already have a direct file URL, skip to the step 3.

# Prepare URL for `Get Presigned URL` API call
$query = "https://api.pdf.co/v1/file/upload/get-presigned-url?
contenttype=application/octet-stream&name=" + `
[System.IO.Path]::GetFileName($SourceFile)
```

```

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

try {
    # Execute request
    $jsonResponse = Invoke-RestMethod -Method Get -Headers @{ "x-api-key" = $API_KEY
} -Uri $query

    if ($jsonResponse.error -eq $false) {
        # Get URL to use for the file upload
        $uploadUrl = $jsonResponse.presignedUrl
        # Get URL of uploaded file to use with later API calls
        $uploadedFileUrl = $jsonResponse.url

        # 2. UPLOAD THE FILE TO CLOUD.

        $r = Invoke-WebRequest -Method Put -Headers @{ "x-api-key" = $API_KEY;
"content-type" = "application/octet-stream" } -InFile $SourceFile -Uri $uploadUrl

        if ($r.StatusCode -eq 200) {

            # 3. MAKE UPLOADED PDF FILE SEARCHABLE

            # Prepare URL for `Make Searchable PDF` API call
            $query = "https://api.pdf.co/v1/pdf/makesearchable?name={0}&password=
{1}&pages={2}&lang={3}&url={4}" -f `
                $(Split-Path $DestinationFile -Leaf), $Password, $Pages, $Language,
$uploadedFileUrl
            $query = [System.Uri]::EscapeUriString($query)

            # Execute request
            $jsonResponse = Invoke-RestMethod -Method Get -Headers @{ "x-api-key" =
$API_KEY } -Uri $query

            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
$DestinationFile -Uri $resultFileUrl

                Write-Host "Generated PDF file saved as `"$($DestinationFile)`"
file."
            }
            else {
                # Display service reported error
                Write-Host $jsonResponse.message
            }
        }
        else {
            # Display request error status
            Write-Host $r.StatusCode + " " + $r.StatusDescription
        }
    }
    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 "&
.\MakeSearchablePdfFromUploadedFile.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](#)