

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

## How to convert CSV to PDF from URL asynchronously for CSV to PDF API in PowerShell: How To Tutorial

The coding tutorials are designed to help you test the features without need to write your own code. PDF.co Web API was made to help with CSV to PDF API in PowerShell. PDF.co Web API is the Rest API that provides set of data extraction functions, tools for documents manipulation, splitting and merging of pdf files. Includes built-in OCR, images recognition, can generate and read barcodes from images, scans and pdf.

This rich sample source code in PowerShell for PDF.co Web API includes the number of functions and options you should do calling the API to implement CSV to PDF API. Open your PowerShell project and simply copy & paste the code and then run your app! Tutorials are available along with installed PDF.co Web API if you'd like to dive deeper into the topic and the details of the API.

Our website provides free trial version of PDF.co Web API that includes source code samples to help with your PowerShell project.

PowerShell - ConvertCsvToPdfFromUrlAsynchronously.ps1

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

# Prepare URL for `CSV To PDF` API call
$query = "https://api.pdf.co/v1/pdf/convert/from/csv?name=$(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 "&
.\ConvertCsvToPdfFromUrlAsynchronously.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](#)