

[www.bytescout.com](http://www.bytescout.com)

## PDF to CSV API in Python using PDF.co Web API

What is PDF.co Web API? It 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.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about PDF.co Web API](#)

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

## Source Code Files:

ConvertPdfToCSVFromUploadedFile.py

```
import os
import requests # pip install requests

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

# Base URL for PDF.co Web API requests
BASE_URL = "https://api.pdf.co/v1"

# Source PDF file
SourceFile = ".\\sample-rotated.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 = ""
# Destination CSV file name
```

```

DestinationFile = ".\\result.csv"

# Some of advanced options available through profiles:
# (JSON can be single/double-quoted and contain comments.)
# {
#   "profiles": [
#     {
#       "profile1": {
#         "CSVSeparatorSymbol": ",", // Separator symbol.
#         "CSVQuotaionSymbol": "\"", // Quotation symbol.
#         "ExtractInvisibleText": true, // Invisible text extraction. Values: true / false
#         "ExtractShadowLikeText": true, // Shadow-like text extraction. Values: true / false
#         "LineGroupingMode": "None", // Values: "None", "GroupByRows", "GroupByColumns", "JoinOrphanedCells"
#         "ColumnDetectionMode": "ContentGroupsAndBorders", // Values: "ContentGroupsAndBorders", "ContentGroupsAndBordersAndText"
#         "Unwrap": false, // Unwrap grouped text in table cells. Values: true / false
#         "ShrinkMultipleSpaces": false, // Shrink multiple spaces in table cells that affect column detection.
#         "DetectNewColumnBySpacesRatio": 1, // Spacing ratio that affects column detection.
#         "CustomExtractionColumns": [ 0, 50, 150, 200, 250, 300 ], // Explicitly specify columns coordinates.
#         "CheckPermissions": true, // Ignore document permissions. Values: true / false
#       }
#     }
#   ]
# }

# Advanced Options
Profiles = "{ 'profiles': [ { 'profile1': { 'RotationAngle': 1 } } ] }"

def main(args = None):
    uploadedFileUrl = uploadFile(SourceFile)
    if (uploadedFileUrl != None):
        convertPdfToCSV(uploadedFileUrl, DestinationFile)

def convertPdfToCSV(uploadedFileUrl, destinationFile):
    """Converts PDF To CSV using PDF.co Web API"""

    # Prepare URL for 'PDF To CSV' API request
    url = "{}/pdf/convert/to/csv?name={}&password={}&pages={}&url={}&profiles={}".format(
        BASE_URL,
        os.path.basename(destinationFile),
        Password,
        Pages,
        uploadedFileUrl,
        Profiles
    )

    # Execute request and get response as JSON
    response = requests.get(url, headers={"x-api-key": API_KEY, "content-type": "application/octet-stream" })
    if (response.status_code == 200):
        json = response.json()

        if json["error"] == False:
            # Get URL of result file
            resultFileUrl = json["url"]
            # Download result file
            r = requests.get(resultFileUrl, stream=True)
            if (r.status_code == 200):
                with open(destinationFile, 'wb') as file:
                    for chunk in r:
                        file.write(chunk)
                print(f"Result file saved as \"{destinationFile}\" file.")
            else:
                print(f"Request error: {response.status_code} {response.reason}")
        else:
            # Show service reported error
            print(json["message"])
    else:
        print(f"Request error: {response.status_code} {response.reason}")

def uploadFile(fileName):
    """Uploads file to the cloud"""

    # 1. RETRIEVE PRESIGNED URL TO UPLOAD FILE.

    # Prepare URL for 'Get Presigned URL' API request
    url = "{}/file/upload/get-presigned-url?contenttype=application/octet-stream&name={}".format(
        BASE_URL, os.path.basename(fileName))

```

```
# Execute request and get response as JSON
response = requests.get(url, headers={"x-api-key": API_KEY })
if (response.status_code == 200):
    json = response.json()

    if json["error"] == False:
        # URL to use for file upload
        uploadUrl = json["presignedUrl"]
        # URL for future reference
        uploadedFileUrl = json["url"]

        # 2. UPLOAD FILE TO CLOUD.
        with open(fileName, 'rb') as file:
            requests.put(uploadUrl, data=file, headers={"x-api-key": API_KEY, "content-type": "application/octet-stream"})

        return uploadedFileUrl
    else:
        # Show service reported error
        print(json["message"])
    else:
        print(f"Request error: {response.status_code} {response.reason}")

return None

if __name__ == '__main__':
    main()
```

---

## VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

## ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit PDF.co Web API Home Page](#)  
[Explore PDF.co Web API Documentation](#)  
[Explore Samples](#)  
[Sign Up for PDF.co Web API 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)

