

How to convert PDF to excel from uploaded file for PDF to excel API in Python and ByteScout Cloud API Server

Learn to write code convert PDF to excel from uploaded file for PDF to excel API in Python: Simple How To Tutorial

The documentation is written to assist you to apply all the necessary features on your side. ByteScout Cloud API Server helps with PDF to excel API in Python. ByteScout Cloud API Server 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..

If you want to speed up the application's code writing then Python code samples for Python developers help to implement using ByteScout Cloud API Server. Follow the tutorial and copy - paste code for Python into your project's code editor. Use of ByteScout Cloud API Server in Python is also described in the documentation given along with the product.

Free! Free! Free! ByteScout free trial version is available for FREE download from our website. Programming tutorials along with source code samples are assembled.

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:

ConvertPdfToExcelFromUploadedFile.py

```
import os
import requests # pip install requests

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

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

# 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 = ""
# Destination Excel file name
DestinationFile = ".\\result.xlsx"

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

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

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

    # Execute request and get response as JSON
    response = requests.get(url, headers={ "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)
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={ "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 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)

