

How to merge PDF documents from uploaded files for PDF merging API in Python with ByteScout Cloud API Server

How to merge PDF documents from uploaded files for PDF merging API in Python: Step By Step Instructions

This page displays the step-by-step instructions and algorithm of how to merge PDF documents from uploaded files and how to apply it in your application. ByteScout Cloud API Server helps with PDF merging API in Python. ByteScout Cloud API Server is the ready to use Web API Server that can be deployed in less than 30 minutes into your own in-house server or into private cloud server. Can store data on in-house local server based storage or in Amazon AWS S3 bucket. Processing data solely on the server using built-in ByteScout powered engine, no cloud services are used to process your data!.

The SDK samples displayed below explain how to quickly make your application do PDF merging API in Python with the help of ByteScout Cloud API Server. Follow the tutorial and copy - paste code for Python into your project's code editor. Enjoy writing a code with ready-to-use sample Python codes to add PDF merging API functions using ByteScout Cloud API Server in Python.

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:

MergePDFDocumentsFromUploadedFile.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 files
SourceFile_1 = "\\sample1.pdf"
SourceFile_2 = "\\sample2.pdf"

# Destination PDF file name
DestinationFile = "\\result.pdf"

def main(args = None):
    UploadedFileUrl_1 = uploadFile(SourceFile_1)
    UploadedFileUrl_2 = uploadFile(SourceFile_2)

    if (UploadedFileUrl_1 != None and UploadedFileUrl_2 != None):
        uploadedFileUrls = "{},{}".format(UploadedFileUrl_1, UploadedFileUrl_2)
        mergeFiles(uploadedFileUrls, DestinationFile)

def mergeFiles(uploadedFileUrls, destinationFile):
    """Perform Merge using PDF.co Web API"""

    # Prepare URL for 'Merge PDF' API request
    url = "{}pdf/merge?name={}&url={}".format(
        BASE_URL,
        os.path.basename(destinationFile),
        uploadedFileUrls
    )

    # 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)

