

How to add text and images to PDF in Python using ByteScout Cloud API Server

Learn to add text and images to PDF in Python

ByteScout simple and easy to understand tutorials are planned to describe the code for both Python beginners and advanced programmers. Want to add text and images to PDF in your Python app? ByteScout Cloud API Server is designed for it. 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..

Want to save time? You will save a lot of time on writing and testing code as you may just take the Python code from ByteScout Cloud API Server for add text and images to PDF below and use it in your application. Just copy and paste the code into your Python application's code and follow the instructions. Enjoy writing a code with ready-to-use sample Python codes.

You can download free trial version of ByteScout Cloud API Server from our website with this and other source code samples for Python.

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:

AddImageByFindingTargetCoordinates.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"

# Direct URL of source PDF file.
SourceFileUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-edit/sample.pdf"

# Search string.
SearchString = 'Your Company Name'

# 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 PDF file name
DestinationFile = "../result.pdf"

# Image params
Type = "image"
Width = 119
Height = 32
ImageUrl = "https://bytescout-com.s3.amazonaws.com/files/demo-files/cloud-api/pdf-edit/logo.png"

def main(args = None):
    # First of all try to find Text within input PDF file
    res = findTextWithinPDF(SourceFileUrl, SearchString)

    if res:
        addImageToPDF(DestinationFile, res['top'], res['left'])
    else:
        print("No result found!")

def findTextWithinPDF(sourceFile, searchText):
    # Prepare URL for PDF text search API call
    # See documentation: https://app.pdf.co/documentation/api/1.0/pdf/find.html

    retVal = dict()

    url = "{}/pdf/find?url={}&searchString={}".format(
        BASE_URL,
        sourceFile,
        searchText
    )

    # 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:
            # print(json)
            if json["body"]:
                retVal['top'] = json["body"][0]['top']
                retVal['left'] = json["body"][0]['left']
            else:
                # Show service reported error
                print(json["message"])
        else:
            print(f"Request error: {response.status_code} {response.reason}")
```

```

return retVal

def addImageToPDF(destinationFile, top, left):
    """Add image using PDF.co Web API"""

    # Prepare URL for 'PDF Edit' API request
    url = "{}pdf/edit/add?name={}&password={}&pages={}&url={}&type={}&x={}&y={}&width={}&height={}&urlimage={}".format(
        BASE_URL,
        os.path.basename(destinationFile),
        Password,
        Pages,
        SourceFileUrl,
        Type,
        top + 300,
        left,
        Width,
        Height,
        ImageUrl
    )

    # 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)
            with open(destinationFile, 'wb') as file:
                for chunk in r:
                    file.write(chunk)
            print(f"Result file saved as '{destinationFile}' file.")
        else:
            # Show service reported error
            print(json["message"])
        else:
            print(f"Request error: {response.status_code} {response.reason}")

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 \[www.PDF.co\]\(http://www.PDF.co\)](#)

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