www.bytescout.com

How to PDF text search API in Python with ByteScout Cloud API Server

If you want to learn more then this tutorial will show how to PDF text search API in Python

The code displayed below will guide you to install an Python app to PDF text search API. 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. and you can use it to PDF text search API with Python.

These Python code samples for Python guide developers to speed up coding of the application when using ByteScout Cloud API Server. This Python sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Check Python sample code samples to see if they respond to your needs and requirements for the project.

ByteScout provides the free trial version of ByteScout Cloud API Server along with the documentation and source code samples.

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

Source Code Files:

PdfTextSearchFromUploadedFile.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 = ""

# Search string.
SearchString = "\d{1,}\.\d\d" # Regular expression to find numbers like '100.00'
                              # Note: do not use `+` char in regex, but use `{1,}` instead.
                              # `+` char is valid for URL and will not be escaped, and it will become a space char on the server s

# Enable regular expressions (Regex)
RegexSearch = True


def main(args = None):
    uploadedFileUrl = uploadFile(SourceFile)
    if (uploadedFileUrl != None):
        searchTextInPDF(uploadedFileUrl)


def searchTextInPDF(uploadedFileUrl):
    """Search Text using PDF.co Web API"""

    # Prepare URL for 'PDF Text Search' API request
    url = "{}/pdf/find?password={}&pages={}&url={}&searchString={}&regexSearch={}".format(
        BASE_URL,
        Password,
        Pages,
        uploadedFileUrl,
        SearchString,
        RegexSearch
    )

    # 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:
            # Display found information
            for item in json["body"]:
                print(f"Found text {item['text']} at coordinates {item['left']}, {item['top']}")
        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

visit www.PDF.co

www.bytescout.com