

How to replace text from PDF in Python with PDF.co Web API

PDF.co Web API 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:

ReplaceTextFromUrlAsynchronously.py

```
""" Cloud API asynchronous "PDF To Text" job example.
    Allows to avoid timeout errors when processing huge or scanned PDF documents.
"""
import os
import requests # pip install requests
import time
import datetime

# 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"

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

```

# PDF document password. Leave empty for unprotected documents.
Password = ""
# Destination PDF file name
DestinationFile = ".\result.pdf"
# (!) Make asynchronous job
Async = True

def main(args = None):
    replaceStringFromPdf(SourceFileURL, DestinationFile)

def replaceStringFromPdf(uploadedFileUrl, destinationFile):
    """Replace Text from PDF using PDF.co Web API"""

    # Prepare URL for 'Replace Text from PDF' API request
    url = "{}/pdf/edit/replace-text?async={}&name={}&password={}&url={}&searchString=The most conspicuous feature
    BASE_URL,
    Async,
    os.path.basename(destinationFile),
    Password,
    uploadedFileUrl
    )

    # 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:
            # Asynchronous job ID
            jobId = json["jobId"]
            # URL of the result file
            resultFileUrl = json["url"]

            # Check the job status in a loop.
            # If you don't want to pause the main thread you can rework the code
            # to use a separate thread for the status checking and completion.
            while True:
                status = checkJobStatus(jobId) # Possible statuses: "working", "failed", "aborted", "success".

                # Display timestamp and status (for demo purposes)
                print(datetime.datetime.now().strftime("%H:%M:%S") + " : " + status)

                if status == "success":
                    # 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}")
                            break
                    elif status == "working":
                        # Pause for a few seconds
                        time.sleep(3)
                    else:
                        print(status)
                        break
                else:
                    # Show service reported error
                    print(json["message"])
            else:
                print(f"Request error: {response.status_code} {response.reason}")

def checkJobStatus(jobId):
    """Checks server job status"""

    url = f"{BASE_URL}/job/check?jobid={jobId}"

    response = requests.get(url, headers={ "x-api-key": API_KEY })
    if (response.status_code == 200):
        json = response.json()
        return json["status"]
    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)

www.bytescout.com