How to convert web page to PDF from link asynchronously for HTML to PDF API in Python with ByteScout Cloud API Server

How to convert web page to PDF from link asynchronously in Python with easy ByteScout code samples to make HTML to PDF API. Step-by-step tutorial

The sample source codes on this page will show you how to create HTML to PDF API in Python. ByteScout Cloud API Server was designed to assist HTML to PDF API in Python. ByteScout Cloud API Server is the ready to deploy Web API Server that can be deployed in less than thirty minutes into your own in-house Windows server (no Internet connnection is required to process data!) 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!.

Use the code displayed below in your application to save a lot of time on writing and testing code. For implementation of this functionality, please copy and paste the code below into your app using code editor. Then compile and run your app. Further improvement of the code will make it more robust.

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

Source Code Files:

ConvertWebPageToPdfFromLinkAsynchronously.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

# 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"

# URL of web page to convert to PDF document.
SourceUrl = "http://www.usa.gov"
# Destination PDF file name
DestinationFile = ".\\result.pdf"
# (!) Make asynchronous job
Async = True


def main(args = None):
    convertLinkToPDF(SourceUrl, DestinationFile)


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

    # Prepare URL for 'URL To PDF' API request
    url = "{}/pdf/convert/from/url?async={}&name={}&url={}".format(
        BASE_URL,
        Async,
        os.path.basename(destinationFile),
        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:
            # 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)
    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 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