Displaying a PDF file on Plotly Dash dashboard in python

kapital1 **Jan 2020**

0

I am working on a simple dashboard using Dash Plotly. On my dashboard, I would like to display a PDF file that is found from my local disc. But I couldn't find a way to do that.

I was trying by converting the PDF file into an image and displaying it as an image file. But my idea is to preview as the PDF file as it is.

Below shown how I tried to approach it.

```
html.Div([
   dcc.Graph(id='pdf_view')
], style= {'width': '50%', 'display': 'inline-block'
```

Here is my function to display the PDF on the browser.

```
def open_pdf (file_path):
     read_pdf = webbrowser.open_new(file_path)
     return read_pdf
```

Here is my callback

```
@app.callback(
Output('pdf_view', 'figure'),
[Input('first-process', 'value')])

def pdf_viewer(one_process):
       return open_pdf(pdf_file)
```

Can anyone help me to do this?

RenaudLN **Jan 2020**

The only way I see to do this with Dash would be to use an iFrame, have you tried this? I'm not sure what your `webbrowser.open_new` returns but there should be a way to embed the pdf.

kapital1 **Jan 2020**

Hi, @RenaudLN Thank you for your reply. I never tried that. Now I am looking at that. I found this. `html.Iframe(src='/assets/mydoc.pdf')` Could you give me a bit of an example of how to display the PDF within the dash app?

Skip to main content

kapital1 **Jan 2020**

Update:

This how I am trying with Iframe. But got an error.

```
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output

app = JupyterDash('sdsds')

app.layout = html.Div(
    [
        html.H1('Title'),
        html.Iframe(src='example.PDF')
    ])
app
```

When I tried this, I got the following error

Unable to understand view name of example.PDF with args {} and app path app/endpoints/8b98bde63956472d941a1619de18f945

Could you help me how to do that?

RenaudLN **Jan 2020**

Based on the answer from **this SO post** , you should use the html `object` tag. In Dash this would look like:

```
app.layout = html.Div(
    [
        html.H1('Title'),
        html.ObjectEl(
            # To my recollection you need to put you
            data="assets/example.PDF",
            type="application/pdf",
            style={"width": "800px", "height": "600p
        ),
    ])
```

Share

kapital1 **Jan 2020**

Thank you @RenaudLN . When I try the script that you gave me, I got the following error:

```
TypeError: Unexpected keyword argument `src`
Allowed arguments: accessKey, aria-*, children, clas
```

And when I try to change the keyword `src` there is no error but the PDF is not displaying.

Topics
More
Categories
Dash Python
Plotly Python
plotly.js
All categories
Tags
announcements
community-components
job-postings
question
show-and-tell
tips-and-tricks
All tags
Skip to main content

Could you help me to approach it?

RenaudLN **Jan 2020**

Can you try with `data` instead of `src`? It was a typo, I edited it.

---

kapital1 **Jan 2020**

When I use data I got the same error. `TypeError: Unexpected keyword argument data`

EDITED

AND it suggests to use `data-*` but this one new feature in HTML5 and I think it is not implemented yet for Dash app.

Could you suggest me an alternative solution?

RenaudLN **Jan 2020**

Alright I actually tried and the iframe solution works. Have you tried it with putting the pdf in the `assets` folder?

```
app.layout = html.Div(
    html.Iframe(id="embedded-pdf", src="assets/examp
)
```

---

kapital1 **Jan 2020**

Yes, I created `assets` folder and I just put the file `example.pdf`. When I use your script I got the following error.

`Unable to understand view name of assets with args {} and app path app/endpoints/caa8bfc3a7184dd7b180feb499e47498`

FYI: I am using `jupyter notebook`.

I am wondering, If it works for you, it could have work for me as well.

Here is my full script.

```
import dash
import dash_html_components as html
import base64

app = JupyterDash('aa')

app.layout = html.Div(
    html.Iframe(id="embedded-pdf", src="assets/examp
)
```

Skip to main content

```
app
```

gage-russell **Apr 2020**

Here is a full working example of how to upload and display PDF’s in a Dash App:

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 21 09:59:11 2020

@author: GRussell
"""

import base64
import datetime
import io
import time

import dash
from dash.dependencies import Input, Output, State
import dash_core_components as dcc
import dash_html_components as html
from pdf2image import convert_from_path, convert_frc

import pandas as pd

from pdf2image import convert_from_bytes

external_stylesheets = ['https://codepen.io/chriddyp

app = dash.Dash(__name__, external_stylesheets=exter

app.layout = html.Div(children=[
    html.Div(
        children = [
            dcc.Upload(
                className="four columns",
                id='upload-coa',
                children=html.Div([
                    'Drag and Drop or ',
                    html.A('Select PDF')
                ]),
                style={
                    'width': '45%',
                    'height': '60px',
                    'lineHeight': '60px',
                    'borderWidth': '1px',
                    'borderStyle': 'dashed',
                    'borderRadius': '5px',
                    'textAlign': 'center',
                    'margin': '10px'
                },
                # Allow multiple files to be uploade
                multiple=False
```

Skip to main content

```
            ),
        ]
    ),
    html.Hr(),
    html.Div(id='output-coa'),
])

def pil_to_b64_dash(im):
    buffered = io.BytesIO()
    im.save(buffered, format="JPEG")
    img_str = base64.b64encode(buffered.getvalue())
    return bytes("data:image/jpeg;base64,", encoding

def parse_coa_contents(contents, filename, date):
    content_type, content_string = contents.split(',

    decoded = base64.b64decode(content_string)

    images = convert_from_bytes(decoded)

    encoded = pil_to_b64_dash(images[0])

    return html.Div([
        # HTML images accept base64 encoded strings
        # that is supplied by the upload
        html.Img(src=encoded.decode('utf-8')),
        html.Hr(),
    ])


@app.callback(Output('output-coa', 'children'),
              [Input('upload-coa', 'contents')],
              [State('upload-coa', 'filename'),
               State('upload-coa', 'last_modified')]
def show_coa(list_of_contents, list_of_names, list_c
    if list_of_contents is not None:
        children = [parse_coa_contents(list_of_conte
        return children

if __name__ == '__main__':
    app.run_server(debug=True)
```

Please excuse any unnecessary imports or code blocks, this is copy and pasted code from another project (I tried to trim it down to just what is necessary for the PDF portion … but I am impatient).

Let me know if there are any questions.

Skip to main content

Related Topics

Topic	Replies	Views	Activity
How download a pdf in my Dash & Plotly application	5	2.4k	Jul 2023
Download Dash HTML page as pdf file?	2	725	Jun 2020
How to load a plotly html file saved locally to dash and serve	2	1.2k	Apr 2021
Displaying generated PDF (not local or external)	6	3.9k	Aug 2022
Including html plotly graphs in dash app	2	3.0k	Feb 2020