Python flask : Standard Templates

Listing file structure

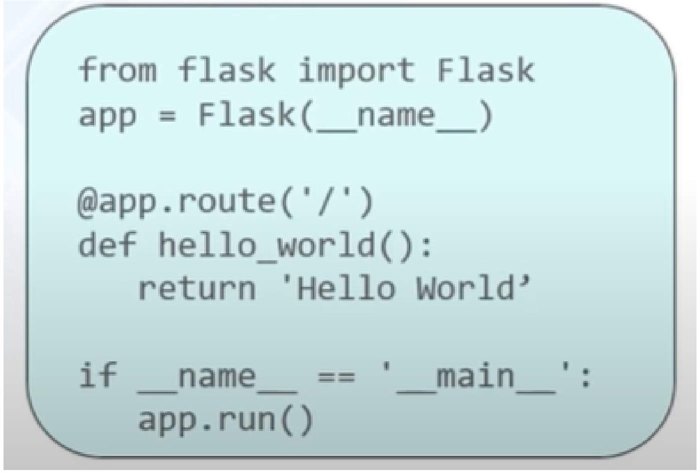
▼ Code

Original link : https://stackoverflow.com/questions/23718236/python-flask-browsing-through-directory-with-files

▼ Python code :

Basic flask with "hello world"

▼ Code



```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
   return 'Hello World’

if __name__ == '__main__':
   app.run()
```

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
  return 'Hello World'
if __name__ == '__main__':
  app.run(host='0.0.0.0')
#If we want to run locally and not on production
#then keep the following line above
#app.run()
```

Basic code to fetch data from link without "?"(question mark): link /<name>

▼ Code

```
from flask import Flask
from flask import request
app = Flask(__name__)

@app.route('/')
def hello_world():
   return 'Hello World'
@app.route('/hello',methods=['GET'])
def hello():
   name1 = request.args.get('name',None)
   # print()
   return 'welcome to my blog page %s' % name
@app.route('/welcome/<name>')
def welcome(name):
   return '<h1>welcome my my blog projedct %s</h1>' %name

if __name__ == '__main__':
   app.run(debug=True,port=8000)
```

Flask for returning json file as response

▼ Code

```
from flask import json
from flask import Flask
app = Flask(__name__)
@app.route('/summary')
def summary():
    contents = open('sample.json', "r").read()
    data = [json.loads(str(item)) for item in contents.strip().split('\n')]
    response = app.response_class(
        response=json.dumps(data),
        mimetype='application/json'
    )
    return response
if __name__ == '__main__':
  app.run(host='0.0.0.0')
```

Link for reference : https://stackoverflow.com/questions/13081532/return-json-response-from-flask-view

Regular Interval Tasks

▼ Code for running a background tasks at regular intervals while hosting webpage

```
import time
import atexit
from apscheduler.schedulers.background import BackgroundScheduler
from flask import Flask
app = Flask(__name__)

def print_date_time():
    print(time.strftime("%A, %d. %B %Y %I:%M:%S %p"))
@app.route('/')
def hello_world():
  return 'Hello World'

scheduler = BackgroundScheduler()
scheduler.add_job(func=print_date_time, trigger="interval", seconds=60)
scheduler.start()

# Shut down the scheduler when exiting the app
atexit.register(lambda: scheduler.shutdown())
if __name__ == '__main__':
  app.run(host='0.0.0.0')
```

Flask for loading json File - INCLUDING webpage template

▼ Code

Python code

```
from flask import json
from flask import Flask
from flask import render_template
app = Flask(__name__)
@app.route('/summary')
def summary():
    contents = open('sample.json', "r").read()
    data = [json.loads(str(item)) for item in contents.strip().split('\n')]
#    print(data[:20])
    #response = app.response_class(
     #   response=json.dumps(data),
     #   mimetype='application/json'
    #)
    return render_template('recordsJson.html',len=len(data) , records=data)
#    return response
if __name__ == '__main__':
        app.run(host='0.0.0.0',port = 5020)
```

Inside the same folder create a folder 'templates' (Very important)

Inside the templates folder , create recordsJson.html

```
<html>
<head>
    <title>Urban Kisaan</title>
</head>
<body>

<ol>
<!-- JSON Data -->
```

```
{%for i in range(0, len)%}

    <li>{{records[i]}}</li>
{%endfor%}

</ol>

</body>
</html>
```

For further reference : https://www.geeksforgeeks.org/python-using-for-loop-in-flask/

Flask for checking arguments from the link itself

▼ Code

```
from flask import request

@app.route('/summary',methods = ['GET'])
def api_summary_2():
    filename = request.args.get('file', None)
    if filename is None:
        abort(404)
    else:
        return api_summary(filename)
```

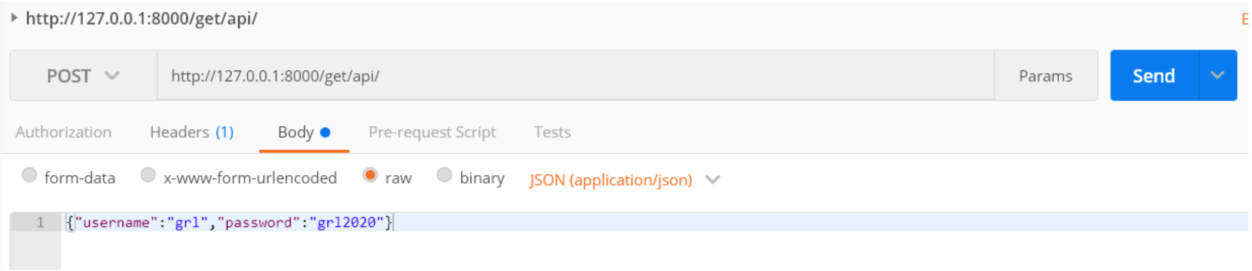
https://stackoverflow.com/questions/13133197/adding-parameter-values-to-a-url-in-flask-python

Flask for recieving arguments from POST Request

▼ Code

```
@app.route('/get/questions/', methods=['GET', 'POST','DELETE', 'PATCH'])
def question():
  if request.method == 'POST':
          average_time = request.form.get('average_time')
          choices = request.form.get('choices')
          created_by = request.form.get('created_by')
          difficulty_level = request.form.get('difficulty_level')
          question = request.form.get('question')
          topics = request.form.get('topics')
#similarly we can write for request.methos=='GET'
```

Flask recieving JSON data from POST (Included postman)



http://127.0.0.1:8000/get/api/

POST http://127.0.0.1:8000/get/api/

```
{"username":"grl","password":"grl2020"}
```

▼ Code

```
from flask import Flask
from flask import request
app = Flask(__name__)
@app.route('/get/api/', methods=['GET', 'POST','DELETE', 'PATCH'])
def question():
    if request.method == 'POST':
        json_data = request.get_json()
        print(json_data['username'])
        return "We got it boys"
if __name__ == '__main__':
    app.run(port=8000)
```

Flask for loading image

▼ Code

```
from flask import Flask, render_template, request,send_file
from werkzeug.utils import secure_filename
from werkzeug.datastructures import  FileStorage
#from nnProcessorThread import NNProcesor
import os
app = Flask(__name__)
@app.route('/upload')
def upload_file():
   return render_template('fileupload3.html')

@app.route('/uploader', methods = ['GET', 'POST'])
def uploader_file():
   if request.method == 'POST':
      f = request.files['file']
      #finalImageName
      f.save(secure_filename(f.filename))
      finalImageName = secure_filename(f.filename)
 #nnProc = NNProcesor(img_file = f.filename)
#      nnProc.start()
  #    finalImageName = nnProc.generatePredictedOutput()
      print(finalImageName)
      #render_template('finalOutput.html',filename = finalImageDir)
      #print(os.getcwd()+'/static/rect_20210812T112955.png')
      return render_template('finalOutput.html',filename = '/'+finalImageName)
@app.route('/', defaults={'req_path': ''})
@app.route('/<path:req_path>')
```

```
def dir_listing(req_path):
    BASE_DIR = os.getcwd()

    # Joining the base and the requested path
    abs_path = os.path.join(BASE_DIR, req_path)

    # Return 404 if path doesn't exist
    if not os.path.exists(abs_path):
        return abort(404)

from flask import Flask, render_template, request,send_file
from werkzeug.utils import secure_filename
from werkzeug.datastructures import  FileStorage
#from nnProcessorThread import NNProcesor
import os
app = Flask(__name__)
@app.route('/upload')
def upload_file():
   return render_template('fileUpload.html')

@app.route('/uploader', methods = ['GET', 'POST'])
def uploader_file():
   if request.method == 'POST':
      f = request.files['file']
      #finalImageName
      f.save(secure_filename(f.filename))
      finalImageName = secure_filename(f.filename)
 #nnProc = NNProcesor(img_file = f.filename)
#      nnProc.start()
  #    finalImageName = nnProc.generatePredictedOutput()
      print(finalImageName)
      #render_template('finalOutput.html',filename = finalImageDir)
      #print(os.getcwd()+'/static/rect_20210812T112955.png')
      return render_template('finalOutput.html',filename = '/'+finalImageName)
@app.route('/', defaults={'req_path': ''})
@app.route('/<path:req_path>')
def dir_listing(req_path):
    BASE_DIR = os.getcwd()

    # Joining the base and the requested path
    abs_path = os.path.join(BASE_DIR, req_path)

    # Return 404 if path doesn't exist
    if not os.path.exists(abs_path):
        return abort(404)
```

▼ HTML code : fileUpload.html

```
<html>
   <body>
      <form action = "http://51.222.234.21:7890/uploader" method = "POST"
         enctype = "multipart/form-data">
         <input type = "file" name = "file" />
         <input type = "submit"/>
      </form>
   </body>
</html>
```

▼ Displaying image : finalOutput.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Index</title>
</head>
<body>
    <img src="{{filename}}" alt="User Image">
</body>
</html>
```

Flask for loading image v2(Proper tutorial online)

▼

Error

▼ TypeError: generatecode() takes 0 positional arguments but 1 was given .Why do we get this error when we are running an outside function within a class?

https://stackoverflow.com/questions/43839536/typeerror-generatecode-takes-0-positional-arguments-but-1-was-given/43839602

This occurs because python by default sends an instance of this class to the next function

Browsing files and directorys

```
@app.route('/files', defaults={'folder': None,'sub_folder': None}, methods=['GET'])
@app.route('/files/<folder>', defaults={'sub_folder': None}, methods=['GET'])
@app.route('/files/<folder>/<sub_folder>', methods=['GET'])

    def files(folder,sub_folder):
        basedir = 'files/'
        directory = ''

        if folder != None:
            directory = directory + '/' + folder

        if sub_folder != None:
            directory = directory + '/' + sub_folder

        files = os.listdir(basedir + directory)

        return render_template('files.html',files=files,directory=basedir + directory,currdir=directory)
```

```
<body>
    <h2>Files {{ currdir }}</h2> </br>
    {% for name in files: %}
        <A HREF="{{ directory }}{{ name }}">{{ name }}</A> </br></br>
    {% endfor %}
</body>s.html',files=files,directory=basedir + directory,currdir=directory)
```

Running flask on google colab

Link : https://www.geeksforgeeks.org/how-to-run-flask-app-on-google-colab/

```
!pip install flask
!pip install flask-ngrok
```

```
from flask import Flask
from flask_ngrok import run_with_ngrok
app = Flask(__name__)
run_with_ngrok(app)

@app.route("/")
def home():
    return "<h1>GFG is great platform to learn</h1>"

app.run()
```

Gunicorn Python basic Template