Flask Workshop

What is Flask?

* Flask is a micro web framework written in Python. It’s a micro framework as it dosent require specific set of tools for itself and can be connected to a database to make a webapp.
* Flask makes use of libraries like Werkzeug and Jinja in order to create applications.

Why use Flask?

* built-in development server and fast debugger
* integrated support for unit testing
* RESTful request dispatching
* Jinja2 templating
* support for secure cookies (client side sessions)
* WSGI 1.0 compliant
* Unicode based

Pre-requisites

So do we directly start coding in flask? No, we have to deal with some concepts before or during the development process.

1. HTTP – HTTP is the most common protocol. We see it every time when we access anything on the internet. But we need to understand what actually happens. Basically, if two devices want to communicate then one must request and the other must return the data. Let’s say we want to access a website. So in this case our device will be the client. The client sends a request to the server that it wants to access the website. The server sends whether the client can access the data or not. The client receives the response and carries out next process.

There are many instructions sent by the client, but we are interested in the http request methods, mainly the GET and POST method. The GET method is used to get information or data from the form. POST is used when we need to get data and we want to change the value in a database.

HTTP concepts include (as the Hypertext part of the name implies) the idea that files can contain references to other files whose selection will elicit additional transfer requests. Any Web server machine contains, in addition to the Web page files it can serve, an HTTP daemon, a program that is designed to wait for HTTP requests and handle them when they arrive. Your Web browser is an HTTP client, sending requests to server machines. When the browser user enters file requests by either "opening" a Web file (typing in a Uniform Resource Locator or URL) or clicking on a hypertext link, the browser builds an HTTP request and sends it to the Internet Protocol address (IP address) indicated by the URL. The HTTP daemon in the destination server machine receives the request and sends back the requested file or files associated with the request. (A Web page often consists of more than one file.)

2. API – API stands for Application Program Interface. So what is an API? API allow two software to communicate with each other. Let’s say you are at a restaurant and you want to order something. Do you go into the kitchen and tell the chef to make you this particular dish? Of course no. You don’t go to the kitchen every time to tell what you want. There comes a waiter who helps you deliver the message to the chef. The waiter acts as an API.

3. JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

JSON is syntax for storing and exchanging data.

My First Flask App

We will be creating a basic weather forecast app using flask framework. Use ide like pycharm, idle or atom.

Let’s start:

import requests

from flask import Flask, render\_template, request

app = Flask(\_\_name\_\_)

app.debug = True

• Firstly start any idea of your choice and create project and a folder called app.py in it

• Import the flask libraries like Flask, requests, render

For our app to know we are implementing it using flask we write

app=flask(\_\_name\_\_)

@app.route('/',methods=['GET','POST'])  
def index():  
  
 if request.method == 'GET':  
  
 city = request.args.get('city')  
  
 if city=='null' or city=='':  
 city='mumbai'

* For every app to work it needs to get info from a server and return the data asked for .All this process are done inside a **function** in python and its defined using **def** keyword.
* Flask has server on which our app runs the IP of that server is passed using @app.route
* We make use of methods like get to receive the data and post to send it to desired server.

**In function**

* The app has passed city for which we wants data and hence **requests** the data from server using API.
* And the name of the city or place to be searched is stored in **city.**
* **If** no city entered (at the start of server) then city searched for ins Mumbai by server.

req = requests.get('http://api.openweathermap.org/data/2.5/weather?q='+city+'&units=imperial&appid=271d1234d3f497eed5b1d80a07b3fcd1').json()

* The info of the city’s weather we wanted is given by this api and all this information about weather, temperature, description is stored in an object called req.
* This a way of collecting the **pre-existing** data using a **web API.**
* data={  
   'city':city,  
   'temperature': req['main']['temp'],  
   'desc': req['weather'][0]['description'],  
   'icon': req['weather'][0]['icon'],  
  }

**Now the data that we stored is in form of json format or in easier way in the form of matrix**

* So to request the data we want we point to location where the particular data is and store the info in different named object for the reference in the future.
* This way of extracting the data is termed as **MAPPING**.

print(data)  
  
 return render\_template('index.html', data=data)  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run()

**NOW all the data once fetched we need to display it on the server**

* The layout in which our data is seen is done using **html** (frontend)
* We need to create a html file to view our data on server and once file is ready the data is passed onto it using render\_template(‘filename’, data)
* **if \_\_name\_\_ == '\_\_main\_\_':** says that if the module is main (name of module)then run the flask app.