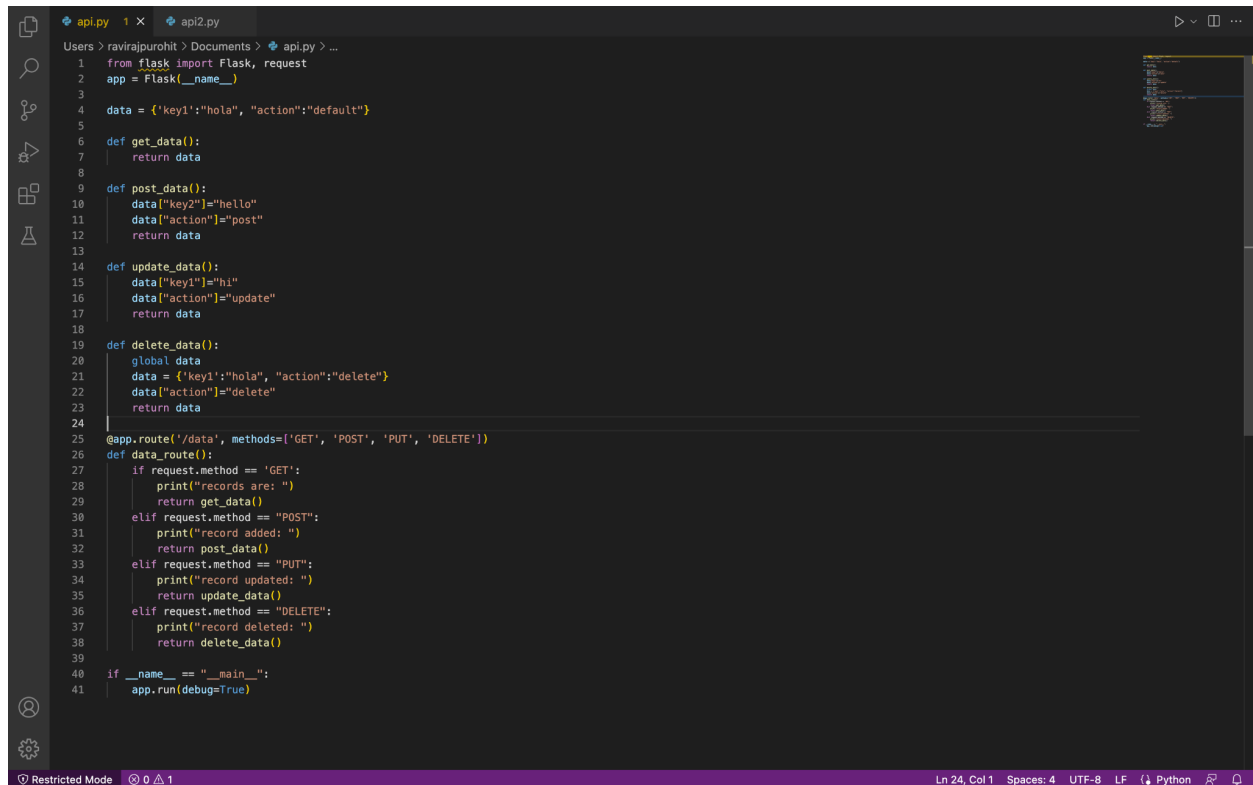


Cloud Computing In Class Lab 2

Name - Ravi Rajpurohit

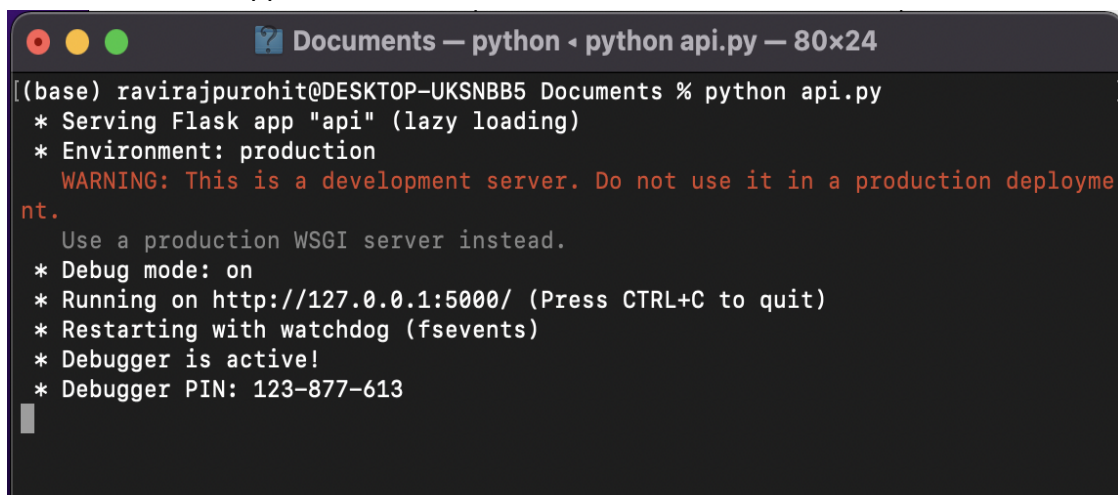
UTA ID - 1002079916

Step 1 - Implement a Flask API with GET, POST, PUT (UPDATE) and DELETE methods
Using Python, defined a simple flask app to store some data in a local dictionary



```
api.py 1 x  api2.py
Users > ravirajpurohit > Documents > api.py > ...
1  from flask import Flask, request
2  app = Flask(__name__)
3
4  data = {'key1': 'hola', 'action': 'default'}
5
6  def get_data():
7      return data
8
9  def post_data():
10     data['key2'] = 'hello'
11     data['action'] = 'post'
12     return data
13
14  def update_data():
15     data['key1'] = 'hi'
16     data['action'] = 'update'
17     return data
18
19  def delete_data():
20     global data
21     data = {'key1': 'hola', 'action': 'delete'}
22     data['action'] = 'delete'
23     return data
24
25  @app.route('/data', methods=['GET', 'POST', 'PUT', 'DELETE'])
26  def data_route():
27     if request.method == 'GET':
28         print("records are: ")
29         return get_data()
30     elif request.method == 'POST':
31         print("record added: ")
32         return post_data()
33     elif request.method == 'PUT':
34         print("record updated: ")
35         return update_data()
36     elif request.method == 'DELETE':
37         print("record deleted: ")
38         return delete_data()
39
40  if __name__ == "__main__":
41     app.run(debug=True)
```

Step 2 - Run the flask application



```
Documents — python < python api.py — 80x24
((base) ravirajpurohit@DESKTOP-UKSNBB5 Documents % python api.py)
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with watchdog (fsevents)
* Debugger is active!
* Debugger PIN: 123-877-613
```

Step 3 - Test the GET method

GET http://127.0.0.1:5000/di

No Environment

http://127.0.0.1:5000/data

Save

</>

GET

http://127.0.0.1:5000/data

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (4)

Test Results

200 OK

11 ms

191 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  
2  "action": "default",  
3  "key1": "hola"  
4
```

Step 4 - Test the POST method

POST http://127.0.0.1:5000/c

No Environment

http://127.0.0.1:5000/data

Save

</>

POST

http://127.0.0.1:5000/data

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (4)

Test Results

200 OK

5 ms

208 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  
2  "action": "post",  
3  "key1": "hola",  
4  "key2": "hello"  
5
```

Step 5 - Test the PUT/UPDATE method

PUT http://127.0.0.1:5000/di + ... No Environment

http://127.0.0.1:5000/data Save

PUT http://127.0.0.1:5000/data Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results 200 OK 5 ms 208 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "action": "update",  
3   "key1": "hi",  
4   "key2": "hello"  
5 }
```

Step 6 - Test the DELETE method

DEL http://127.0.0.1:5000/di + ... No Environment

http://127.0.0.1:5000/data Save

DELETE http://127.0.0.1:5000/data Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results 200 OK 4 ms 190 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "action": "delete",  
3   "key1": "hola"  
4 }
```