

## FOXMULA ASSIGNMENT

Creating a user in sqlite database and fetching the user based on id using rest APIs

**Technologies used :** Python, Flask and SQLite database. Flask is a lightweight framework for web applications using python.

Importing libraries and setting up the app :

```
import flask
from flask import request, jsonify
from flask_restful import Resource, Api

import sqlite3

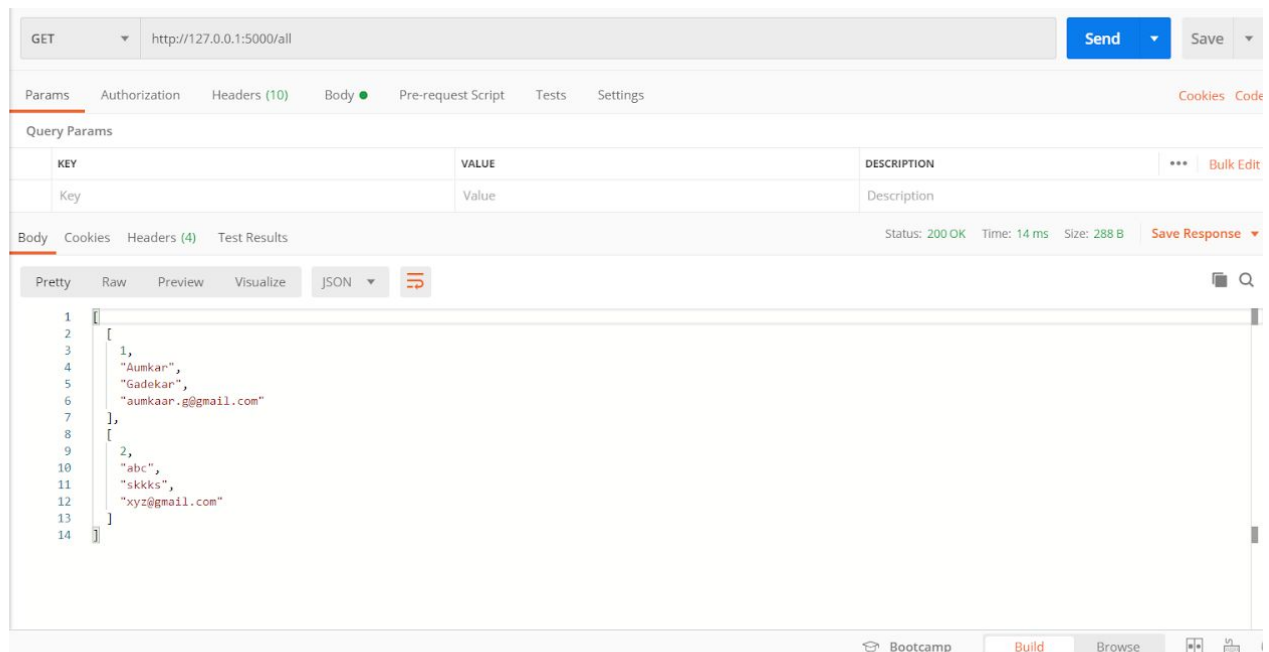
app = flask.Flask(__name__)
api=Api(app)
```

The database was created using SQLite and populated with 2 entries. Four columns - id, first name, last name and email describe the user.

```
sqlite> select * from user;
1|Aumkar|Gadekar|aumkaar.g@gmail.com
2|abc|skkks|xyz@gmail.com
```

GET API for displaying all users :

```
class all(Resource):
    def get(self):
        conn = sqlite3.connect('users.db')
        cur = conn.cursor()
        all_users= cur.execute('SELECT * FROM user;').fetchall()
        return jsonify(all_users)
```



GET API for displaying user based on id entered in parameter :

```
class byid(Resource):
    def get(self, id):
        query = "SELECT * FROM user WHERE"
        to_filter = []
        if id:
            query += ' id=? AND'
            to_filter.append(id)
        if not (id):
            return page_not_found(404)

        query = query[:-4] + ';'
        conn = sqlite3.connect('users.db')
        cur = conn.cursor()
        results = cur.execute(query, to_filter).fetchall()
        return jsonify(results)
```

GET http://127.0.0.1:5000/userbyid/1

Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 11 ms Size: 223 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   [
3     1,
4     "Aumkar",
5     "Gadekar",
6     "aumkaar.g@gmail.com"
7   ]
8 ]
```

```
@app.route('/userbyid', methods=['GET'])
def api_filter():
    query_parameters = request.args

    id = query_parameters.get('id')
    print(id)
    query = "SELECT * FROM user WHERE id=" + id + ";"
    conn = sqlite3.connect('users.db')
    cur = conn.cursor()
    result = cur.execute(query).fetchall()
    if len(result) == 0:
        return jsonify({"error": "ID NOT FOUND"})
    return jsonify(result)
```

GET http://127.0.0.1:5000/userbyid?id=2

Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	2	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 47 ms Size: 212 B

Pretty Raw Preview Visualize JSON

```
1 [
2   [
3     2,
4     "abc",
5     "skkks",
6     "xyz@gmail.com"
7   ]
8 ]
```

Error message when ID doesn't exist :

The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: http://127.0.0.1:5000/userbyid?id=5
- Send button: Blue button with white text.
- Params: 1 param (id=5)
- Authorization: None
- Headers (10): None
- Body: None
- Pre-request Script: None
- Tests: None
- Settings: None
- Status: 200 OK
- Time: 9 ms
- Size: 176 B
- Response Body (JSON):

```
{
  "error": "ID NOT FOUND"
}
```

Create new user with unique ID and return it :

```
@app.route("/create", methods=['POST'])
def create():
    if request.method == 'POST':
        fname = request.json['fname']
        lname = request.json['lname']
        email = request.json['email']

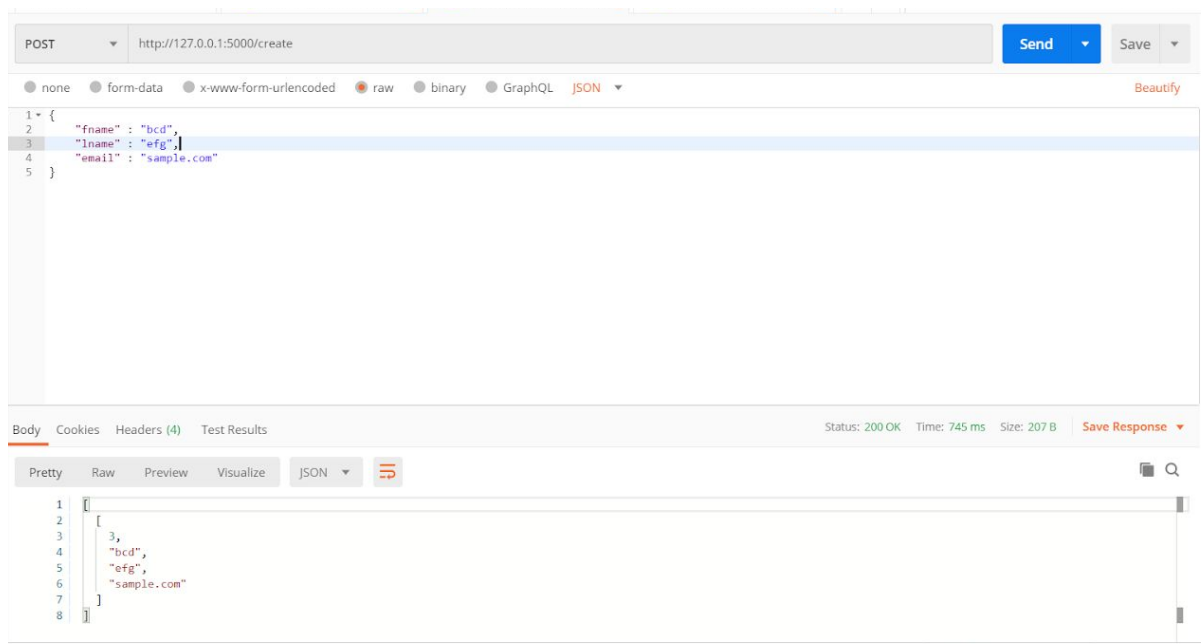
        try:
            conn = sqlite3.connect('users.db')
            cur = conn.cursor()
            sql='SELECT * FROM user WHERE email = "' +email+'";'
            print(sql)
            email_match= cur.execute(sql).fetchall()
            if(email_match):
                return jsonify({'error':'email exists'})
            num=cur.execute("select num from info").fetchall()
            new_id=num[0][0]+1
            print(new_id) #Generates
new ID

            sql="update info set num = "+str(new_id)+";"
            cur = conn.cursor()
            cur.execute(sql)
            conn.commit()
```

```

cur.execute( "INSERT INTO user(id,fname,lname,email)
VALUES(?, ?, ?, ?);", (new_id,fname,lname,email))
conn.commit()
query = "SELECT * FROM user WHERE id=" +str(new_id)+";"
result = cur.execute(query).fetchall()
return jsonify(result)
except Exception as e:
    print(e)
    return "Exception encountered"
return page_not_found

```



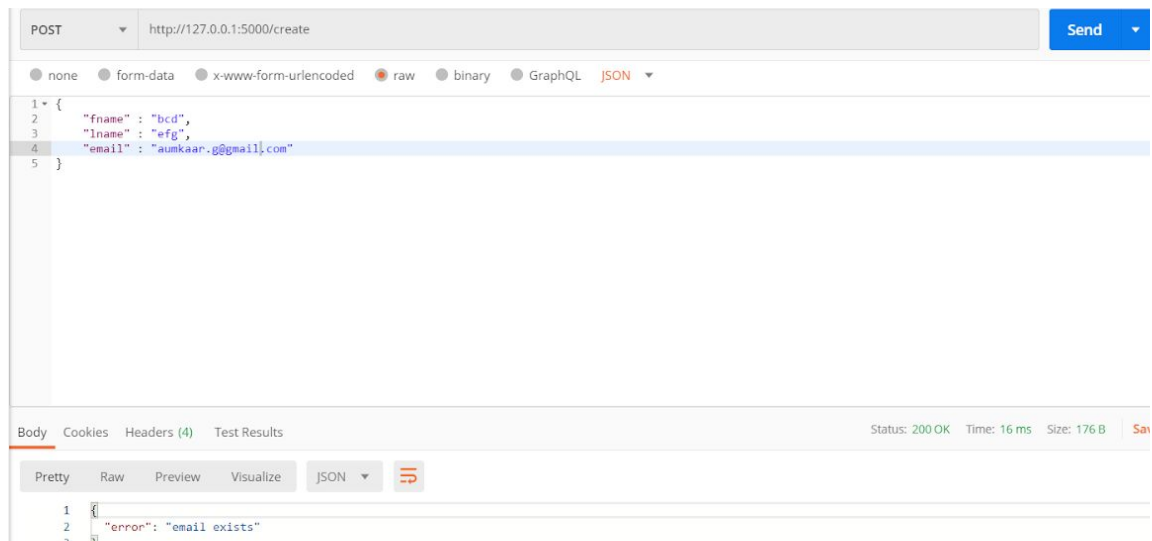
Updated database after adding :

```

sqlite> select * from user;
1|Aumkar|Gadekar|aumkaar.g@gmail.com
2|abc|skkks|xyz@gmail.com
sqlite> select * from user;
1|Aumkar|Gadekar|aumkaar.g@gmail.com
2|abc|skkks|xyz@gmail.com
3|bcd|efg|sample.com
sqlite>

```

In case email already exists :



Add paths and run the app :

```
api.add_resource(byid, '/userbyid/<id>')
api.add_resource(all, '/all')
```

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