SQL API using Python

We will use Flask for this API

```python
from flask import Flask, render_template, request, jsonify
import log_py
import sql
import mongodb
import casandra
app = Flask(__name__)

@app.route('/sql_postman', methods=['POST'])  # for calling the API from Postma
def sql_via_postman():
    if (request.method == 'POST'):
        '''
        JSON format
        {
            "operation":"create",
            "host":"host url",
            "user":"username",
            "password":"password",
            "db":"database name",
            "table":"table name",
        }
        '''
        operation = request.json['operation']
        host = request.json['host']
        user = request.json['user']
        password = request.json['password']
        db = request.json['db']
        ob=sql.sql(host,user,password,db)
        table_name=request.json['table']
        print(host,user, password, db, table_name)
```

```
if (operation == 'create'):
    '''
    for creating table
    JSON format
    {
        "columns":{"column name":"datatype(size)",...."}
    }
    '''
    col = request.json['columns']
    columns = ""
    for i in col:
        columns += i + " " + col[i] + ","
    columns=columns[:-1]
    ob.create_table(table_name, columns)
    msg = "Table created"
elif (operation == 'insert'):
    '''
    for creating table
    JSON format
    {
        "data":"data sepreated by comma"
    }
    '''
    data  = request.json['data']
    print(f"INSERT INTO {table_name} VALUES ({data})")
    ob.insert(table_name,data)
    msg = "data inserted"

elif (operation == 'update'):
    '''
    for creating table
    JSON format
    {
        "set": "key=value pair of columns & values to be updated"
        "where": "condition"
    }
    '''
    set = request.json['set']
    where= request.json['where']


    ob.update(table_name, set,where)
    msg = "data updated"
```

```
    if (operation == 'bluk'):
        '''
        for dumping csv
        JSON format
        {
            "f_path":"file path"
            "columns":{"column name":"datatype(size)",...."}
        }
        '''
        f_path = request.json['filepath']
        col = request.json['columns']
        columns = ""
        for i in col:
            columns += i + " " + col[i] + ","
        columns = columns[:-1]
        ob.dump_file(f_path, table_name, columns)
        msg = "data dumped"

    if (operation == 'delete'):
        '''
       for deleting table
       JSON format
       {
           "table":"table name",
           "where": "condition"
       }
       '''

        where = request.json['where']

        ob.delete(table_name, where)
        msg = "data deleted"

    if (operation == 'download'):
        '''
           for downloading table
        '''
        link=ob.download(table_name)
        msg = "you can download data using this link :   http://127.0.0.1:5000/" +link

    return jsonify(msg)


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

Sample JSON Input

CREATE TABLE

```
{
 "operation":"create",
 "host":"localhost",
 "user":"root",
 "password":"1234",
 "db":"UCI",
 "table":"api",
 "columns":{"id":"INT(2)","name":"VARCHAR(20)"}
}
```

INSERT INTO TABLE

```
{
 "operation":"insert",
 "host":"localhost",
 "user":"root",
 "password":"2001",
 "db":"UCI",
 "table":"api",
 "data":"1,Arjun"
}
```

UPDATE TABLE

```
{
 "operation":"update",
 "host":"localhost",
 "user":"root",
 "password":"2001",
 "db":"UCI",
 "table":"api",
 "set":"name ='abhay'",
 "where":"id='1'"
}
```

Sample JSON Input

DELETE

```
{
"operation":"delete",
"host":"localhost",
"user":"root",
"password":"2001",
"db":"UCI",
"table":"api",
"where":"id='1'"
}
```

DOWNLOAD

```
{
"operation":"download",
"host":"localhost",
"user":"root",
"password":"2001",
"db":"UCI",
"table":"api"
}
```

BLUK

```
{
"operation":"bluk",
"host":"localhost",
"user":"root",
"password":"2001",
"db":"UCI",
"table":"api1",
"columns":{"id":"INT(2)","name":"VARCHAR(20)"},
"filepath":"api.csv"

}
```

Now You must be thinking where is our sql.py file, We are calling its function for every operation

Last week I posted a SQL with Python guide, Go through it.

If still you face any doubt comment your mail id for full code.
If you have any doubt in flask, check-out my flask API Guide

FOLLOW ME FOR MORE!

Coming up soon

- MongoDb with Python Guide
- MongoDB API

arjun-panwar

www.linkedin.com/in/arjun-panwar