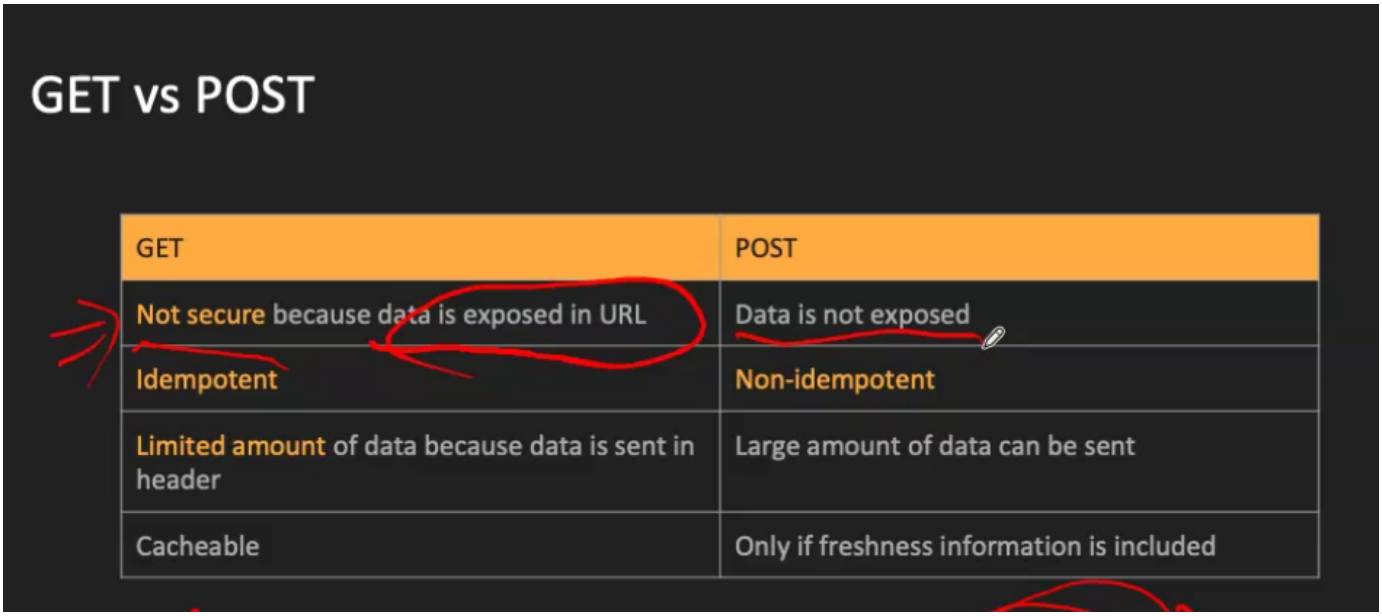
計網5

5/12



proxy:
網路代理，這種網路服務在客戶端和網站之間扮演中介的角色

flask 教學
https://hackmd.io/@shaoeChen/HJiZtEngG/https%3A%2F%2Fhackmd.io%2Fs%2FH1CMTVheG (https://hackmd.io/@shaoeChen/HJiZtEngG/https%3A%2F%2Fhackmd.io%2Fs%2FH1CMTVheG)

Download nginx

```
sudo apt update
sudo apt install nginx

sudo ufw app list #列出結果
sudo ufw allow 80
sudo ufw allow 5000
sudo ufw allow 3306
sudo ufw status #如果是inactive -> sudo ufw enable

sudo systemctl restart nginx
#if got error:
#sudo fuser -k 80/tcp
#sudo fuser -k 443/tcp
#nginx -t see error-> sudo vim /var/log/nginx/error.log
```

參考 (https://phoenixnap.com/kb/how-to-install-nginx-on-ubuntu-20-04)

LAB10

修改 nginx

```
sudo vim /etc/nginx/sites-available/default
```

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        root /home/q56101078/restful_web;

#       server_name 10.100.100.87, 192.168.84.8;
        location / {
                proxy_pass http://localhost:5000/;
        }
```

安裝mysql

```
sudo apt install mysql-server
mysql --version # check mysql was successfully installed
sudo mysql -u root # login mysql as root
```

問題

1. #1698
   before:

```
# --print-defaults to see which it would actually u
+------------------+-----------------------+
| user             | plugin                |
+------------------+-----------------------+
| debian-sys-maint | caching_sha2_password |
| mysql.infoschema | caching_sha2_password |
| mysql.session    | caching_sha2_password |
| mysql.sys        | caching_sha2_password |
| root             | auth_socket           |
+------------------+-----------------------+
5 rows in set (0.00 sec)
```

after:

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select user,host,plugin from mysql.user;
+------------------+-----------+-----------------------+
| user             | host      | plugin                |
+------------------+-----------+-----------------------+
| debian-sys-maint | localhost | caching_sha2_password |
| mysql.infoschema | localhost | caching_sha2_password |
| mysql.session    | localhost | caching_sha2_password |
| mysql.sys        | localhost | caching_sha2_password |
| root             | localhost | mysql_native_password |
+------------------+-----------+-----------------------+
5 rows in set (0.00 sec)
```

`sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf`

```
skip-grant-tables #在[mysqld]增加
#進入mysql
sudo mysql -u root
> use mysql;
> update user set plugin='mysql_native_password' where user='root';
> flush privileges;
> exit;

sudo service mysql restart
```

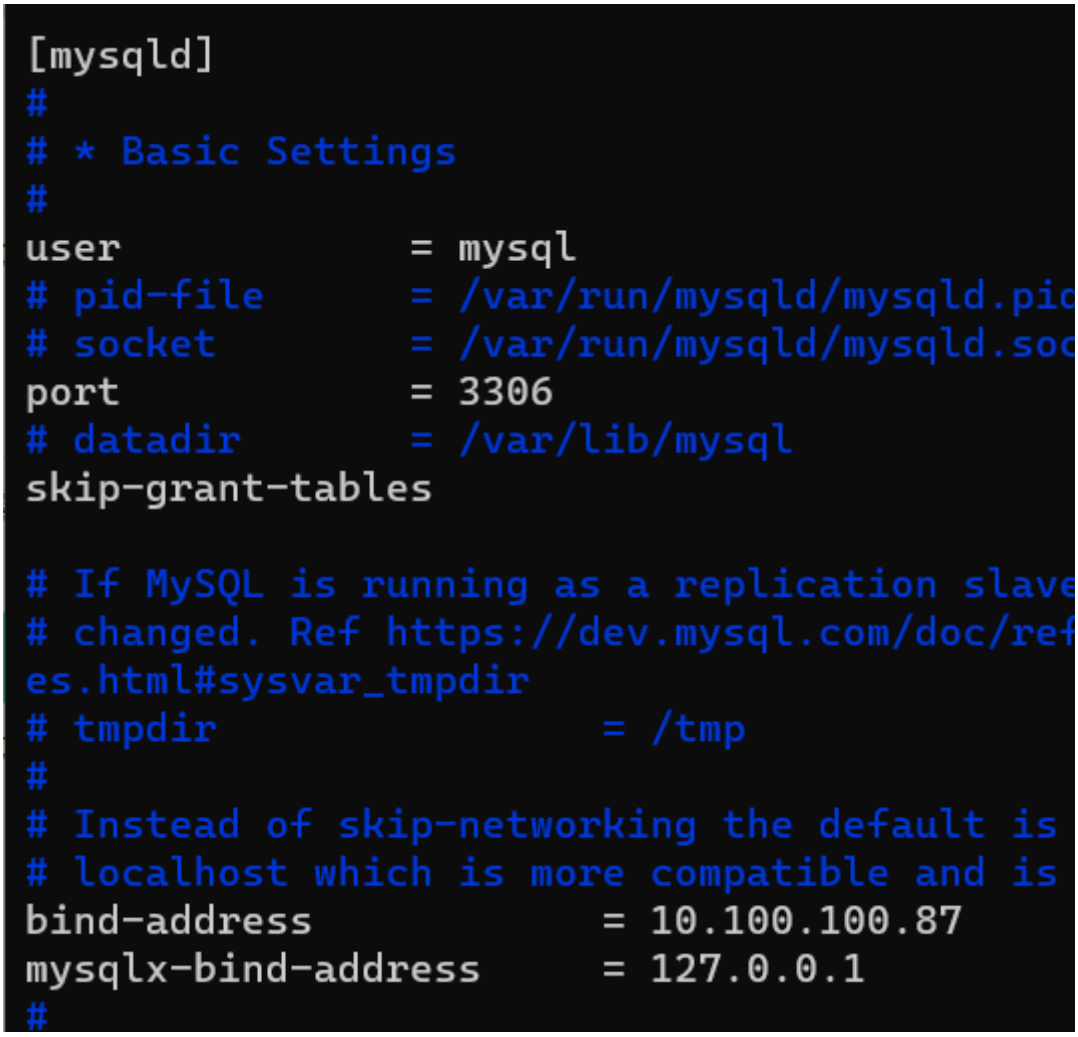
参考1 (https://blog.csdn.net/jlu16/article/details/82809937)

参考2 (https://blog.csdn.net/weixin_35679269/article/details/113419219?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-113419219-blog-113471266.pc_relevant_paycolumn_v3&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-113419219-blog-113471266.pc_relevant_paycolumn_v3&utm_relevant_index=5)

2. #2003

```
sudo vim /etc/mysql/mysql.cnf
sudo vim /etc/mysql/mysql.conf.d/mysqld.conf

sudo systemctl restart mysql
sudo systemctl restart mysqld
```



下面兩行註解調

安裝 python

```
sudo apt-get update
sudo apt-get upgrade
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get install python3.7

python3 main.py # 運行程式
```

參考:Build Simple RESTful API using Python and MySQL (https://techarise.com/restful-api-using-python-mysql/)
參考:Build Simple RESTful API using Python and MySQL2 (https://techarise.com/restful-api-using-python-mysql/)
參考:Flask實作 (https://hackmd.io/@shaoeChen/HJiZtEngG/https%3A%2F%2Fhackmd.io%2Fs%2FBk2QpUAlG)

執行

```
curl -X POST http://10.100.100.87:5000/key -H 'Content-Type: application/json' -
curl -X GET http://10.100.100.87:5000/key -H 'Content-Type: application/json' #2
curl -X GET http://10.100.100.87:5000/key/111q -H 'Content-Type: application/jsc
curl -X PUT http://10.100.100.87:5000/key/111 -H 'Content-Type: application/json

curl -X DELETE http://10.100.100.87:5000/key/111
     -H "Accept: application/json" #5
```

sql 語法

```
desc <table_name>; /*看schema*/

create database <name>;
drop database <name>;
```

需求

1. `POST /key`: 新增一個 key-value pair
    a. Request body 須包含 key 和 value 的值，Ex:

```
{
  "key": "key1",
  "value": "data1"
}
```

    b. 資源成功創建須回傳 201，若 key 已存在則回傳 400 (key 須為唯一)。
2. `GET /key`: 回傳 database 中所有的 key (不用回傳 value)
    a. Response 為 array，包含 database 中所有的 key，Ex:

```
[
  "key1",
  "key2"
]
```

    b. 回傳 200 狀態
3. `GET /key/{key}`: 回傳 database 中指定的 key 的資訊
    a. Request，Ex: `curl http://<YOUR_IP>/key/key1`
    b. Response，Ex:

```
{
  "key1": "data1"
}
```

    c. 成功則回傳 200，若找不到對應的 key 就回傳 404
4. `PUT /key/{key}`: 更新指定的 key 的 value
    a. Request body 須包含 value 的值，Ex:

```
{
  "value": "updated_data1"
}
```

    b. 因為是 PUT，所以 key 不存在時會創建新的 key-value pair，請回傳 201。若 key 已存在，則更新 value 並回傳 200。
5. `DELETE /key/{key}`: 刪除指定的 key
    a. Request，Ex: `curl -X DELETE http://<YOUR_IP>/key/key1`
    b. 回傳 200

RESTFUL API

```
----restful_web
 |--app.py
 |--config.py # mysql 連線方式
 └--main.py #restful api
```

app

```
from flask import Flask
from falsk_cors import CORS,cross_origin

app=Flask(__name__)
CORS(app)
```

config

```
from app import app
from flaskext.mysql import MySQL

mysql = MySQL()
app.config['MYSQL_DATABASE_USER'] = 'root'
app.config['MYSQL_DATABASE_PASSWORD'] = ''
app.config['MYSQL_DATABASE_DB'] = 'restful'
app.config['MYSQL_DATABASE_HOST'] = 'localhost'
mysql.init_app(app)
```

main

```
import pymysql
from app import app
from config import mysql
from flask import jsonify
from flask import Flask,request
import urllib.parse

@app.route("/")
def create_tabel():
    conn=mysql.connect()
    cursor = conn.cursor()
    sql = '''CREATE TABLE lab(`key` VARCHAR(255), `value` VARCHAR(255), PRIMARY KEY (`
    cursor.execute(sql)
    conn.close()
    return "create new table!"

@app.route('/key',methods=['POST','GET'])
def key():
    if request.method == 'POST':
        result = request.get_json()
        key_ = urllib.parse.quote(result['key'],safe='"')
        value_ = urllib.parse.quote(result['value'],safe='"')

        conn = mysql.connect()
        cursor = conn.cursor(pymysql.cursors.DictCursor)
        cursor.execute("SELECT * FROM lab WHERE `key`=%s",key_)
        rows = cursor.fetchall()
        if rows:
            res = jsonify('existed')
            res.status = 400
            return res

        else:
            sql = "INSERT INTO lab(`key`,`value`) VALUES(%s,%s)"
            data = (key_,value_)
            cursor.execute(sql,data)
            conn.commit()
            res = jsonify('insert successfully')
            res.status_code= 201
            return res

    else:
        ans = []
        sql = "SELECT `key` FROM lab"
        conn= mysql.connect()
        cursor = conn.cursor(pymysql.cursors.DictCursor)
        cursor.execute(sql)
        rows = cursor.fetchall()
        for row in rows:
            ans.append(urllib.parse.unquote(row['key']))

        res = jsonify(ans)
```

```
        res.status_code = 200
        return res

@app.route('/key/<path:msg>',methods=['PUT','GET','DELETE'])
def key_fetch(msg):
    if request.method == 'GET':
        msg = urllib.parse.quote(msg,safe='"')
        conn=mysql.connect()
        cursor = conn.cursor(pymysql.cursors.DictCursor)
        cursor.execute("SELECT * FROM lab WHERE `key`=%s",msg)
        row = cursor.fetchall()
        if row:
            ans = {}
            for i in row:
                ans[urllib.parse.unquote(i['key'])] = urllib.parse.unquote(i['value'])
            res = jsonify(ans)
            res.status_code = 200
            return res
        else:
            res=jsonify("not found")
            res.status_code=404
            return res
    elif request.method =='PUT':

        result = request.get_json()
        msg = urllib.parse.quote(msg,safe='"')
        result = urllib.parse.quote(result['value'],safe='"')
        conn=mysql.connect()
        cursor= conn.cursor(pymysql.cursors.DictCursor)
        cursor.execute("SELECT `key` FROM lab WHERE `key`=%s",msg)
        row = cursor.fetchall()
        if row:
            cursor.execute("UPDATE lab set `value`=%s WHERE `key`=%s",(result,msg))
            conn.commit()
            res=jsonify("update successfully")
            res.status_code=200
            return res
        else:
            sql = "INSERT INTO lab(`key`,`value`) VALUES(%s,%s)"
            data = (msg,result)
            cursor.execute(sql,data)
            conn.commit()
            res=jsonify("insert successfully")
            res.status_code=201
            return res

    elif request.method == 'DELETE':
        msg = urllib.parse.quote(msg,safe='"')
        conn = mysql.connect()
        cursor = conn.cursor()
        cursor.execute("DELETE from lab where `key`=%s",msg)
        conn.commit()
        res=jsonify("insert successfully")
```

```
        res.status_code=200

        return res


if __name__ == "__main__":
    app.run(host = "0.0.0.0",debug=True)
```