

網頁前端開發

張家瑋 博士

副教授

國立臺中科技大學資訊工程系



FLASK

H
T
T
P

FLASK

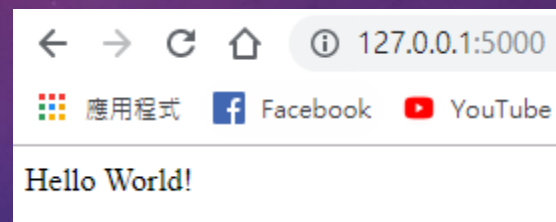
FLASK

- Python 有許多能用於創建 Web 應用程式和 Web API 的框架，而輕量的 Flask 框架可以勝任Web API 的需求：管理 HTTP 請求和顯示資料內容。
- flask-restful 是針對 restful api 開發的一個flask的套件，建構在 flask 的輕薄短小的基礎下，flask-restful 可以在短短幾行內完成 restful api的開發。

第一個 FLASK 範例

```
1 from flask import Flask
2
3 app = Flask(__name__)
4 @app.route("/")
5
6
7 def hello():
8     return "Hello World!"
9
10 if __name__ == "__main__":
11     app.run()
```

```
(venv) C:\Users\user\Desktop\flask>py HelloWorld.py
* Serving Flask app "HelloWorld" (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: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [11/Oct/2019 20:44:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Oct/2019 20:44:52] "GET /favicon.ico HTTP/1.1" 404 -
```



GET DATA BY WEB API

```
1 import flask
2 from flask import jsonify
3
4 app = flask.Flask(__name__)
5 app.config["DEBUG"] = True
6
7 # test data
8 tpe = {
9     "id": 0,
10    "city_name": "Taipei",
11    "country_name": "Taiwan",
12    "is_capital": True,
13    "location": {
14        "longitude": 121.569649,
15        "latitude": 25.036786
16    }
17 }
18 nyc = {}
19 }
20 ldn = {}
21 }
22 cities = [tpe, nyc, ldn]
23
24
25 @app.route('/', methods=['GET'])
26 def home():
27     return "<h1>Hello Flask!</h1>"
28
29
30 @app.route('/cities/all', methods=['GET'])
31 def cities_all():
32     return jsonify(cities)
33
34 app.run()
```

← → ↺ 🏠 ⓘ 127.0.0.1:5000/cities/all

🌐 應用程式 Facebook 📺 YouTube 📺 爱奇艺

```
[
  {
    "city_name": "Taipei",
    "country_name": "Taiwan",
    "id": 0,
    "is_capital": true,
    "location": {
      "latitude": 25.036786,
      "longitude": 121.569649
    }
  },
  {
    "city_name": "New York",
    "country_name": "United States",
    "id": 1,
    "is_capital": false,
    "location": {
      "latitude": 40.710405,
      "longitude": -74.004364
    }
  },
  {
    "city_name": "London",
    "country_name": "United Kingdom",
    "id": 2,
    "is_capital": true,
    "location": {
      "latitude": 51.507497,
      "longitude": -0.114089
    }
  }
]
```

```
(venv) C:\Users\user\Desktop\flask>py GetData.py
* Serving Flask app "GetData" (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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 315-344-113
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [23/Oct/2019 09:32:23] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Oct/2019 09:32:36] "GET /cities/all HTTP/1.1" 200 -
```

GET DATA WITH CHINESE

```
1 import flask
2 from flask import jsonify
3
4 app = flask.Flask(__name__)
5 app.config["DEBUG"] = True
6 app.config["JSON_AS_ASCII"] = False
7
8 # test data
9 tpe = {
10     "id": 0,
11     "city_name": "台北",
12     "country_name": "台灣",
13     "is_capital": True,
14     "location": {
15         "longitude": 121.569649,
16         "latitude": 25.036786
17     }
18 }
19 nyc = {
20 }
21 ldn = {
22 }
23
24 cities = [tpe, nyc, ldn]
25
26 @app.route('/', methods=['GET'])
27 def home():
28     return "<h1>Hello Flask!</h1>"
29
30 @app.route('/cities/all', methods=['GET'])
31 def cities_all():
32     return jsonify(cities)
33
34 app.run()
```

```
[
  {
    "city_name": "台北",
    "country_name": "台灣",
    "id": 0,
    "is_capital": true,
    "location": {
      "latitude": 25.036786,
      "longitude": 121.569649
    }
  },
  {
    "city_name": "紐約",
    "country_name": "美國",
    "id": 1,
    "is_capital": false,
    "location": {
      "latitude": 40.710405,
      "longitude": -74.004364
    }
  },
  {
    "city_name": "倫敦",
    "country_name": "英國",
    "id": 2,
    "is_capital": true,
    "location": {
      "latitude": 51.507497,
      "longitude": -0.114089
    }
  }
]
```

```
(venv) C:\Users\user\Desktop\flask>py GetDatawithChinese.py
* Serving Flask app "GetDatawithChinese" (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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 315-344-113
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [23/Oct/2019 09:41:33] "GET /cities/all HTTP/1.1" 200 -
```

GET DATA BY NAME

from flask import request

```
@app.route('/cities', methods=['GET'])
def city_name():
    if 'city_name' in request.args:
        city_name = request.args['city_name']
    else:
        return "Error: No city_name provided. Please specify a city_name."
    results = []

    for city in cities:
        if city['city_name'] == city_name:
            results.append(city)

    return jsonify(results)
```



The screenshot shows a web browser window with the address bar displaying the URL `127.0.0.1:5000/cities?city_name=台北`. Below the address bar, there are several icons for social media and other services. The main content area of the browser displays a JSON array containing one object representing the city of Taipei.

```
[
  {
    "city_name": "台北",
    "country_name": "台灣",
    "id": 0,
    "is_capital": true,
    "location": {
      "latitude": 25.036786,
      "longitude": 121.569649
    }
  }
]
```

GET DATA FROM CSV

gapminder.csv

	A	B	C	D	E	F
1	country	continent	year	lifeExp	pop	gdpPercap
2	Afghanistan	Asia	1952	28.801	8425333	779.4453
3	Afghanistan	Asia	1957	30.332	9240934	820.853
4	Afghanistan	Asia	1962	31.997	10267083	853.1007
5	Afghanistan	Asia	1967	34.02	11537966	836.1971
6	Afghanistan	Asia	1972	36.088	13079460	739.9811
7	Afghanistan	Asia	1977	38.438	14880372	786.1134
8	Afghanistan	Asia	1982	39.854	12881816	978.0114
9	Afghanistan	Asia	1987	40.822	13867957	852.3959
10	Afghanistan	Asia	1992	41.674	16317921	649.3414
11	Afghanistan	Asia	1997	41.763	22227415	635.3414
12	Afghanistan	Asia	2002	42.129	25268405	726.7341
13	Afghanistan	Asia	2007	43.828	31889923	974.5803
14	Albania	Europe	1952	55.23	1282697	1601.056
15	Albania	Europe	1957	59.28	1476505	1942.284
16	Albania	Europe	1962	64.82	1728137	2312.889
17	Albania	Europe	1967	66.22	1984060	2760.197
18	Albania	Europe	1972	67.69	2263554	3313.422
19	Albania	Europe	1977	68.93	2509048	3533.004
20	Albania	Europe	1982	70.42	2780097	3630.881

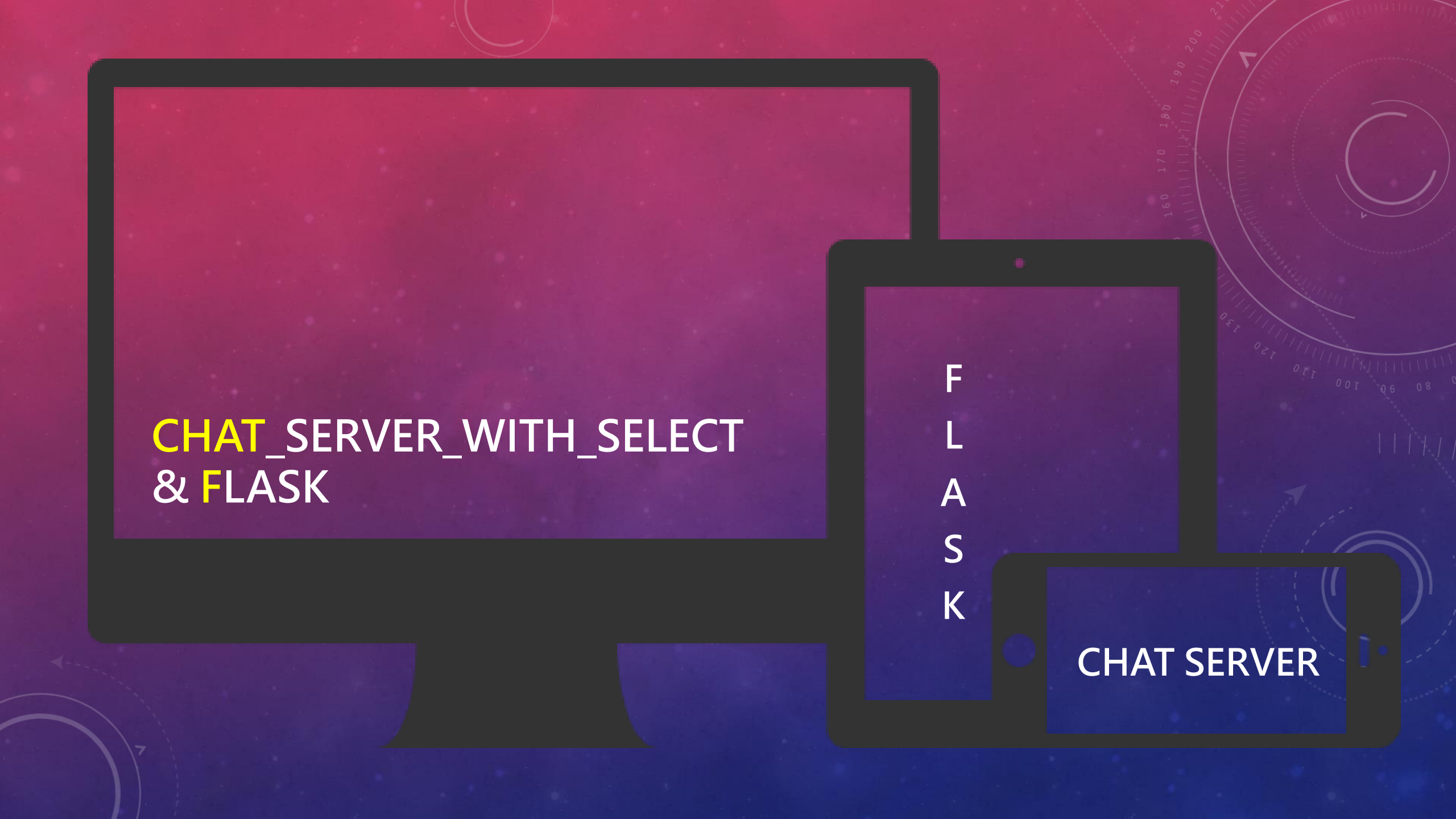
```
127.0.0.1:5000/gapminder?country=Taiwan

[
  {
    "continent": "Asia",
    "country": "Taiwan",
    "gdpPercap": 1206.947913,
    "lifeExp": 58.5,
    "pop": 8550362.0,
    "year": 1952
  },
  {
    "continent": "Asia",
    "country": "Taiwan",
    "gdpPercap": 1507.86129,
    "lifeExp": 62.4,
    "pop": 10164215.0,
    "year": 1957
  },
  {
    "continent": "Asia",
    "country": "Taiwan",
    "gdpPercap": 1822.879028,
    "lifeExp": 65.2,
    "pop": 11918938.0,
    "year": 1962
  },
  {
    "continent": "Asia",
    "country": "Taiwan",
    "gdpPercap": 2643.8586809999997,
    "lifeExp": 67.5,
    "pop": 13648692.0,
    "year": 1967
  },
]
```


GET DATA FROM CSV

```
1 import flask
2 from flask import jsonify, request
3 import numpy as np
4 import pandas as pd
5
6 app = flask.Flask(__name__)
7 app.config["DEBUG"] = True # True 表示開啟除錯模式, 正式對外運行時需註解掉
8 app.config["JSON_AS_ASCII"] = False # False 表示不編譯為 ASCII
9
10 gapminder = pd.read_csv("gapminder.csv")
11 gapminder_list = []
12 nrows = gapminder.shape[0]
13 ▼ for i in range(nrows):
14     ser = gapminder.loc[i, :]
15     row_dict = {}
16 ▼     for idx, val in zip(ser.index, ser.values):
17         if type(val) is str:
18             row_dict[idx] = val
19         elif type(val) is np.int64:
20             row_dict[idx] = int(val)
21         elif type(val) is np.float64:
22             row_dict[idx] = float(val)
23     gapminder_list.append(row_dict)
```

```
31 @app.route('/gapminder/all', methods=['GET'])
32 def gapminder_all():
33     return jsonify(gapminder_list)
34
35
36 @app.route('/gapminder', methods=['GET'])
37 def country():
38     if 'country' in request.args:
39         country = request.args['country']
40     else:
41         return "Error: No country provided. Please specify a country."
42     results = []
43
44     for elem in gapminder_list:
45         if elem['country'] == country:
46             results.append(elem)
47
48     return jsonify(results)
49
50
51 app.run()
```



CHAT_SERVER_WITH_SELECT
& **FLASK**

F
L
A
S
K

CHAT SERVER

安裝所需模組

1. pip install flask
2. pip install flask-cors

```
from flask_cors import *  
CORS(app, resources=r'/*')
```

模組

1. pip install flask

2. pip install flask-socketio



APP.PY

APP.PY

```
1 from flask import Flask, render_template
2 from flask_socketio import SocketIO
3
4 app = Flask(__name__)
5 app.config['SECRET_KEY'] = '1126'
6 socketio = SocketIO(app)
7
8 @app.route('/')
9 def sessions():
10     return render_template('view.html')
11
12 def messageReceived(methods=['GET', 'POST']):
13     print('message was received!!!')
14
15 @socketio.on('my event')
16 def handle_my_custom_event(json, methods=['GET', 'POST']):
17     print('received my event: ' + str(json))
18     socketio.emit('my response', json, callback=messageReceived)
19
20 if __name__ == '__main__':
21     socketio.run(app, debug=True)
```



TEMPLATES/VIEW.HTML

VIEW.HTML

Gary

Messages

提交

Gary: 666

Jason: 7777

Gary: 555

Jason: 444

Gary: 333

Jason: 222

Gary: 1111

Jason

Jason: 7777

Gary: 555

Jason: 444

Gary: 333

Jason: 222

Gary: 1111

命令提示符 - py app.py

127.0.0.1 - - [26/Nov/2019 02:42:38] "GET /socket.io/?EIO=3&transport=polling&t=MwZtscS&sid=d12ad344db9243d0b475d29f825e0951 HTTP/1.1" 200 183 0.001000

127.0.0.1 - - [26/Nov/2019 02:43:18] "GET / HTTP/1.1" 200 1781 0.004008

127.0.0.1 - - [26/Nov/2019 02:43:18] "GET /socket.io/?EIO=3&transport=websocket&sid=d12ad344db9243d0b475d29f825e0951 HTTP/1.1" 200 0 40.393013

127.0.0.1 - - [26/Nov/2019 02:43:18] "GET /socket.io/?EIO=3&transport=polling&t=MwZuOUK HTTP/1.1" 200 349 0.000000

(15020) accepted ('127.0.0.1', 3993)

received my event: {'data': 'User Connected'}

127.0.0.1 - - [26/Nov/2019 02:43:18] "POST /socket.io/?EIO=3&transport=polling&t=MwZu0Us&sid=b875253f445440dda5c855908fd859ec HTTP/1.1" 200 219 0.001993

127.0.0.1 - - [26/Nov/2019 02:43:18] "GET /socket.io/?EIO=3&transport=polling&t=MwZu0Un&sid=b875253f445440dda5c855908fd859ec HTTP/1.1" 200 183 0.004987

received my event: {'user_name': 'Gary', 'message': '666'}

(15020) accepted ('127.0.0.1', 4007)

127.0.0.1 - - [26/Nov/2019 02:45:53] "GET / HTTP/1.1" 200 1616 0.001995

127.0.0.1 - - [26/Nov/2019 02:45:54] "GET /socket.io/?EIO=3&transport=polling&t=MwZucXZ HTTP/1.1" 200 349 0.000000

(15020) accepted ('127.0.0.1', 4010)

received my event: {'data': 'User Connected'}

(15020) accepted ('127.0.0.1', 4011)

127.0.0.1 - - [26/Nov/2019 02:45:54] "POST /socket.io/?EIO=3&transport=polling&t=MwZucXu&sid=f2894e99c020405d824632e8c3abe7ad HTTP/1.1" 200 219 0.001994

127.0.0.1 - - [26/Nov/2019 02:45:54] "GET /socket.io/?EIO=3&transport=polling&t=MwZucXv&sid=f2894e99c020405d824632e8c3abe7ad HTTP/1.1" 200 183 0.001996

received my event: {'user_name': 'Jason', 'message': '7777'}

received my event: {'user_name': 'Gary', 'message': '555'}

received my event: {'user_name': 'Jason', 'message': '444'}

received my event: {'user_name': 'Gary', 'message': '333'}

received my event: {'user_name': 'Jason', 'message': '222'}

received my event: {'user_name': 'Gary', 'message': '1111'}

VIEW.HTML

```
1 <html lang="en">
2 <head>
3   <title>Flask_Chat_App</title>
4 </head>
5 <body>
6
7   <form action="" method="POST">
8     <input type="text" class="username" placeholder="User Name"/>
9     <input type="text" class="message" placeholder="Messages"/>
10    <input type="submit"/>
11  </form>
12
13  <h3 style='color: #ccc;font-size: 30px;'>No message yet..</h3>
14  <div class="message_holder"></div>
15
16  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
17  <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.7.3/socket.io.min.js"></script>
18  <script type="text/javascript">
42  </script>
43
44 </body>
45 </html>
```

VIEW.HTML

```
<script type="text/javascript">
  var socket = io.connect('http://' + document.domain + ':' + location.port);
  socket.on( 'connect', function() {
    socket.emit( 'my event', {
      data: 'User Connected'
    } )
    var form = $( 'form' ).on( 'submit', function( e ) {
      e.preventDefault()
      let user_name = $( 'input.username' ).val()
      let user_input = $( 'input.message' ).val()
      socket.emit( 'my event', {
        user_name : user_name,
        message : user_input
      } )
      $( 'input.message' ).val( '' ).focus()
    } )
  } )
  socket.on( 'my response', function( msg ) {
    console.log( msg )
    if( typeof msg.user_name !== 'undefined' ) {
      $( 'h3' ).remove()
      $( 'div.message_holder' ).append( '<div><b style="color: #000">'+msg.user_name+' : </b> '+msg.message+'</div>' )
    }
  })
</script>
```



THANK YOU