

建立一個完整的flask

建立好以下code之後，記得執行

```
# 載入必須套件
from flask import Flask, request
from flask_restful import Resource, Api

# 創建Flask app物件
app = Flask(__name__)
api = Api(app)

# 創建陣列，用於存放品項
items = [
    {
        "name": "apple",
        "price": 32.3,
    },
    {
        "name": "orange",
        "price": 30,
    },
    {
        "name": "onion",
        "price": 44.5
    }
]

i = [
    {
        "name": "a",
        "num": 1,
    },
    {
        "name": "b",
        "num": 2,
    },
    {
        "name": "c",
        "num": 3
    }
]

class Item(Resource):
```

```

# 單一品項查詢
def get(self, name):
    item = next(filter(lambda x: x['name'] == name, items), None)
    return {'item': item, "i:" i}, 200 if item else 404

# 建制新品項
def post(self, name):
    # 如果該品項已經存在 items 內，就找出並回傳給客戶端該品項已經存在
    if next(filter(lambda x: x['name'] == name, items), None):
        return {'message': f'An item with name {name} already exists ..'}, 403

    # 如果該品項不存在，則解析客戶端傳來的body，並將其品項寫入 items
    data = request.get_json()
    item = {'name': name, 'price': data['price']}
    items.append(item)
    return item, 201

class ItemsList(Resource):
    # 取得所有品項
    def get(self):
        return {'items': items, 'i': i}

api.add_resource(Item, '/item/<string:name>')
api.add_resource(ItemsList, '/items')

if __name__ == "__main__":
    app.run(port=5000, debug=True)

```

注意：class中的功能 "get"、"post"，對應到postman中的功能選項名稱，不能隨便命名