

Random Foodpanda Linebot

隨機點餐機(中央)

組長:109403533/林采璇 組員:109403048/林子恩

專題製作動機

生活在中央大學內，平日最煩惱的不是學業，而是今天要吃甚麼？

宵夜街跟後門都吃膩了，又碰到疫情不想出門，這時我們突然想到為何不做個foodpanda推薦機器人呢？

之後透過推送訊息到Line，每日定時傳送推薦美食，我們從此不必煩惱要吃什麼，隨機點餐機都幫我們想好了！

爬蟲

```
{
  'name': '惡魔炸蛋蔥油餅',
  'payment_types': [],
  'post_code': '320011',
  'primary_cuisine_id': 214,
  'rating': 5.0,
  'redirection_url': 'https://foodpanda.com.tw/resta',
  'review_number': 4,
  'review_with_comment_number': 0,
  'score': 0.0,
  'service_fee_percentage_amount': 0,
  'service_tax_percentage_amount': 0,
  'tag': 'NEXTGEN_NEW_TAG',
  'tags': [{
    'code': 'NEW',
    'text': 'NEXTGEN_NEW_TAG'
  }],
  'url_key': 'e-mo-zha-dan-cong-you-bing',
  'vat_percentage_amount': 0,
  'characteristics': {
    'cuisines': [{
      'id': 214,
      'name': '小吃',
      'url_key': 'twsnacks',
      'main': True
    }],
    'food_characteristics': [],
    'primary_cuisine': {
```

```
def crawlrestaurant(self):
    url = 'https://disco.deliveryhero.io/listing/api/v1/pandora/vendors'
    query = {
        'longitude': 121.1952988, # 中央大學經度
        'latitude': 24.9681558, # 中央大學緯度
        'language_id': 6,
        'include': 'characteristics',
        'dynamic_pricing': 0,
        'configuration': 'Variant1',
        'country': 'tw',
        'budgets': '', # 低中高123
        'cuisine': '',
        'sort': 'rating_desc', # 評分最高
        'food_characteristic': '',
        'use_free_delivery_label': False,
        'vertical': 'restaurants',
        'limit': 50,
        'offset': 0,
        'customer_type': 'regular'
    }
    headers = {
        'x-disco-client-id': 'web',
    }
    r = requests.get(url=url, params=query, headers=headers)

    if r.status_code == requests.codes.ok:
        data = r.json()

        restaurants = data['data']['items']
```

Random

透過random 從50筆資料選出一筆, 內容有餐廳名、評分及該餐廳在foodpanda的連結

```
draw = random.choice(restaurants[:50])
content1 = "抽到" + draw['name'] + "\n" + "評價為" + \
    str(draw['rating']) + "\n" + draw['redirection_url']
# print("抽到" + draw['name'])
# print("排名為" + str(draw['rating']))
# print(draw['redirection_url'])
return content1
else:
    content2 = "請求失敗"
    return content2
# print('請求失敗')
```

定時傳訊與LINE API

- 定時傳訊:
Schedule套件
- Line API:
broadcast

```
50 def morningontimemessage():
51     getfoodontime = Randomcrawlrestaurant()
52     line_bot_api.broadcast(TextSendMessage(
53         text="早安!起床囉!" + '\n' + "這是每日早上7:30定時播送, 現在正在測試中...敬請期待^^" + \
54         '\n' + getfoodontime.crawlrestaurant()))
55
56
57 def noonontimemessage():
58     getfoodontime = Randomcrawlrestaurant()
59     line_bot_api.broadcast(TextSendMessage(
60         text="午安!今天過一半了喔!" + '\n' + "這是每日中午11:30定時播送, 現在正在測試中...敬請期待^^" + \
61         '\n' + getfoodontime.crawlrestaurant()))
62
63
64 def eveningontimemessage():
65     getfoodontime = Randomcrawlrestaurant()
66     line_bot_api.broadcast(TextSendMessage(
67         text="該吃晚餐囉!" + '\n' + "這是每日中午19:00定時播送, 現在正在測試中...敬請期待^^" + \
68         '\n' + getfoodontime.crawlrestaurant()))
69
70
71 schedule.every().day.at("07:30").do(morningontimemessage)
72 schedule.every().day.at("11:30").do(noonontimemessage)
73 schedule.every().day.at("19:00").do(eveningontimemessage)
74 # 測試用
75 # schedule.every(30).seconds.do(ontimemessage)
76
77 while True:
78     schedule.run_pending()
79
```

line robot 建置 Django 框架

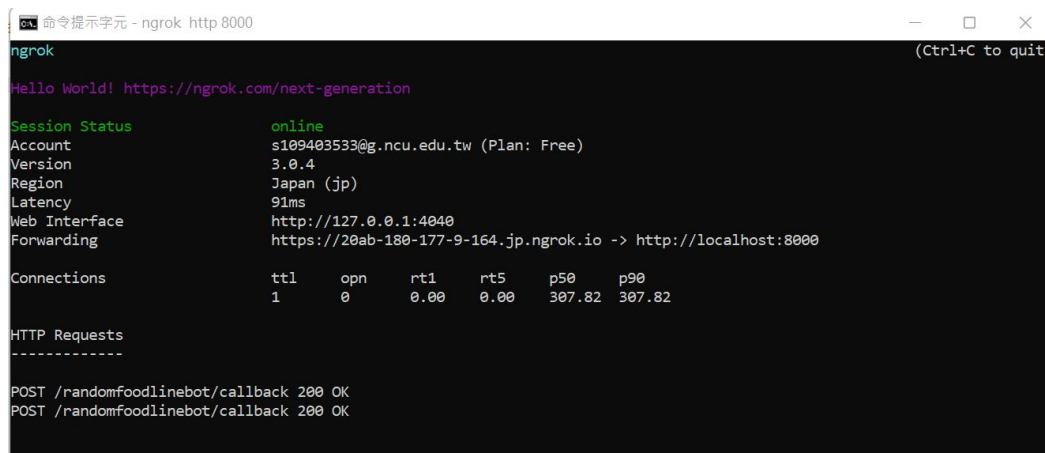
- 為求快速開發, 使用 Django 框架來建置 LINE Bot 應用程式 (APP)
 - 是一個開放原始碼的 Web 應用框架
 - Django 的框架已經定義了專案的基本架構及程式碼撰寫規範
- 以物件導向的方式, 使 APP 可以呼叫爬蟲程式

▽ Django

```
▽ FinalProject_1094035...
  ▽ mylinebot
    ▽ mylinebot
      > __pycache__
      + __init__.py
      + asgi.py
      + settings.py
      + urls.py
      + wsgi.py
    ▽ randomfoodlinebot
      > __pycache__
      > migrations
      + __init__.py
      + admin.py
      + apps.py
      + randomcrawl.py
      + urls.py
      + views.py
    ≡ db.sqlite3
    + manage.py
```

line robot 建置_Ngrok

- 使用Ngrok , 一個轉發的伺服器, 將本機的Port對外公開, 進行傳訊與接收。



```
命令提示字元 - ngrok http 8000

ngrok
Hello World! https://ngrok.com/next-generation

Session Status      online
Account             s109403533@g.ncu.edu.tw (Plan: Free)
Version             3.0.4
Region              Japan (jp)
Latency             91ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://20ab-180-177-9-164.jp.ngrok.io -> http://localhost:8000

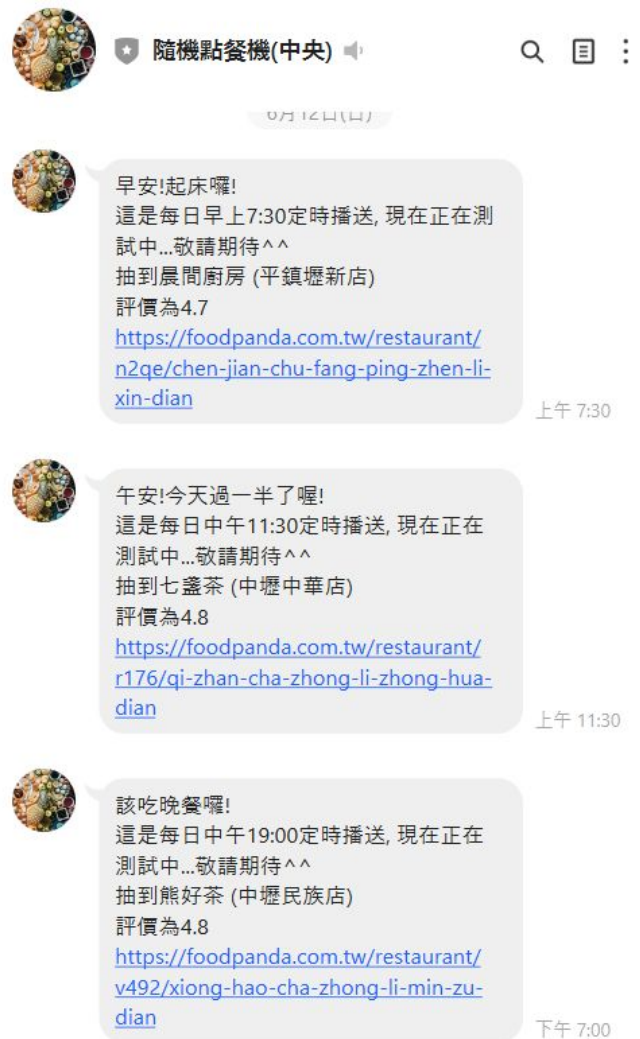
Connections          ttl    opn    rt1    rt5    p50    p90
1                   0     0.00   0.00   307.82 307.82

HTTP Requests
-----
POST /randomfoodlinebot/callback 200 OK
POST /randomfoodlinebot/callback 200 OK
```

△ Ngrok

執行截圖

- 因為我們設定7:30、11:30、19:00推播訊息, 故執行結果如右:



未來展望

目前透過使用者端傳遞訊息並加以回復的功能尚未完成, 或許未來可以加入傳入所在地址、想要食物類別, 也期待之後有時間有想法可以加入新的功能。

今天

已讀
下午 2:31

安安



仍在開發測試中...敬請期待^^
抽到MUMU咖啡 中壢中山店
評價為4.8
<https://foodpanda.com.tw/restaurant/f7ol/mumuka-pei-zhong-li-zhong-shan-dian>

下午 2:31

▼ 使用line API 中的reply_message API 接收並回覆訊息

```
18 # 未來可開發方向，依照傳入訊息回覆
19 # 可以讓使用者輸入所在地區、食物喜好種類等等
20
21 @csrf_exempt
22 def callback(request):
23     if request.method == 'POST':
24         signature = request.META['HTTP_X_LINE_SIGNATURE']
25         body = request.body.decode('utf-8')
26
27         try:
28             events = parser.parse(body, signature) # 傳入的事件
29         except InvalidSignatureError:
30             return HttpResponseForbidden()
31         except LineBotApiError:
32             return HttpResponseBadRequest()
33
34         for event in events:
35             if isinstance(event, MessageEvent): # 如果有訊息事件
36                 getfood = Randomcrawlrestaurant()
37                 line_bot_api.reply_message( # 回復傳入的訊息文字
38                     event.reply_token,
39                     TextSendMessage(text='仍在開發測試中...敬請期待^^' + \
40                                     '\n' + getfood.crawlrestaurant())
41                 )
42                 # TextSendMessage(text=event.message.text)
43             )
44         return HttpResponse()
45     else:
46         return HttpResponseBadRequest()
47
```

Reference

- foodpanda 爬蟲
https://blog.jiatool.com/posts/foodpanda_spider/
- line developer and account manager
<https://developers.line.biz/zh-hant/>
<https://tw.linebiz.com/login/>
- line robot 建置
<https://www.learncodewithmike.com/2020/03/django-install.html>
<https://www.learncodewithmike.com/2020/03/django-create-app.html>
<https://www.learncodewithmike.com/2020/06/python-line-bot.html>
- 定時播送
python schedule
<https://www.readfog.com/a/1648168828734115840>
<https://pypi.org/project/schedule/>
line api overview
<https://github.com/line/line-bot-sdk-python>
<https://developers.line.biz/en/docs/messaging-api/overview/>
line broadcast api
<https://developers.line.biz/en/reference/messaging-api/#send-broadcast-message>
line reply message api
<https://developers.line.biz/en/reference/messaging-api/#send-reply-message>