

國立高雄科技大學智慧商務系

110 學年度專題實驗報告

臺鐵車站即將到站資訊



國立高雄科技大學

National Kaohsiung University of
Science and Technology

指導老師：謝文川教授

學生：C109156127 林駿騰

中華民國 110 年 6 月

目錄

第壹章、緒論	2
壹.第一節 前言	2
壹.第二節 研究動機	2
壹.第三節 研究目的	2
壹.第四節 研究目標	2
壹.第五節 研究流程	3
壹.第六節 系統開發工具	4
壹.第七節 計畫時程	5
壹.第八節 研究範疇與限制	5
第貳章、系統設計與架構	6
貳.第一節 系統架構	6
第參章、系統實作與展示	7
參.第一節 平台的授權設定	7
參.第二節 主要程式展示	7
參.第三節 系統執行結果	8
第肆章、結論與未來展望	9
第伍章、參考文獻	9

第壹章、緒論

壹.第一節 前言

善用公共運輸工具不僅能夠環保，更能夠改善交通，台灣鐵路是大多數人到外縣市的眾多選擇之一。

壹.第二節 研究動機

因就讀高中離我家有一段距離，於是高中時期的我經常運用公車、火車等公共運輸工具，但我認為在 Play 商店公開的應用程式沒有操作方便又能符合我需求的 APP，所以決定嘗試製作有關公共運輸的主題。

壹.第三節 研究目的

在現代化的城市中，複雜的交通讓觀光客以及非本地人經常搞不清楚，希望本次研究能夠減少使用者查詢時刻表的時間。

壹.第四節 研究目標

期盼能讓使用者對搭乘公共運輸工具能夠更加方便。

壹.第五節 研究流程

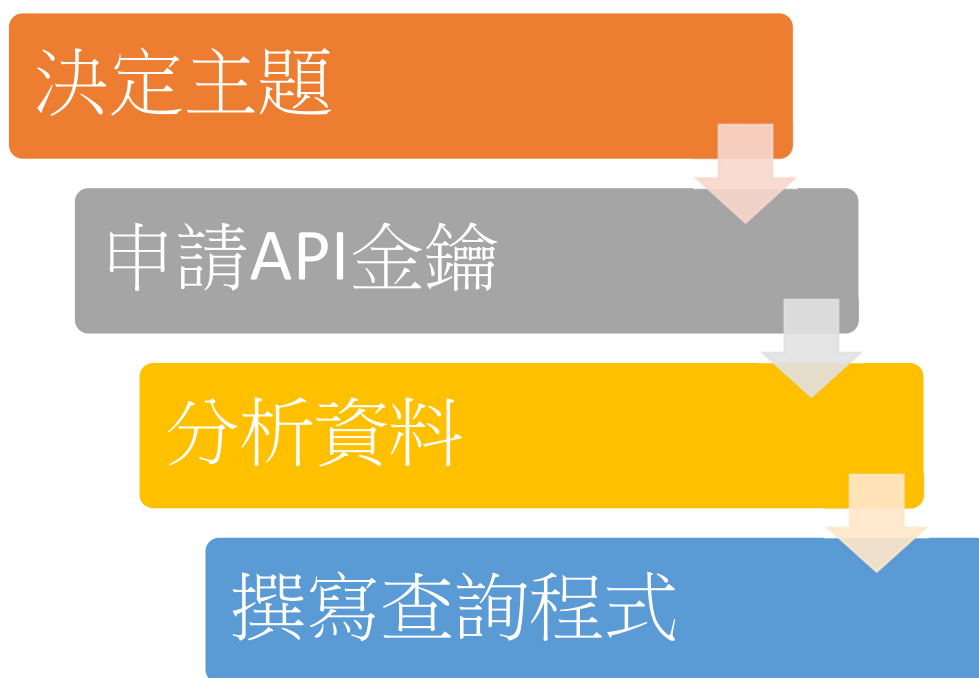


圖 1-1 研究流程圖

1. 決定主題

在政府公開資訊平臺找不到合適的資料，後來找到了公共運輸整合流通平臺。

2. 申請 API 金鑰

註冊會員並申請 API 金鑰。

3. 分析資料

測試平臺上各個 API 的功能，找到合適的資料。

4. 撰寫查詢程式

利用 Visual Studio Code 撰寫程式。

壹.第六節 系統開發工具

1. Visual Studio Code

由 Microsoft 開發且跨平台的免費程式碼編輯器，支援絕大多數的程式語言，並且可以安裝擴充套件增加更多應用。

2. Python 3.7.6

一種被廣泛使用的直譯式高階程式語言，這個程式語言強調可讀性及簡潔的語法，適合初學者學習，但其缺點是執行速度慢，較不適合用於撰寫含有大量程式碼的程式。

3. MOTC Transport API V2

由公共運輸整合資訊流通服務平臺提供的 API，該平臺涵蓋全國公車、臺鐵、高鐵、捷運、航空、自行車等等公共運輸工具的旅運資料服務 API。

壹.第七節 計畫時程

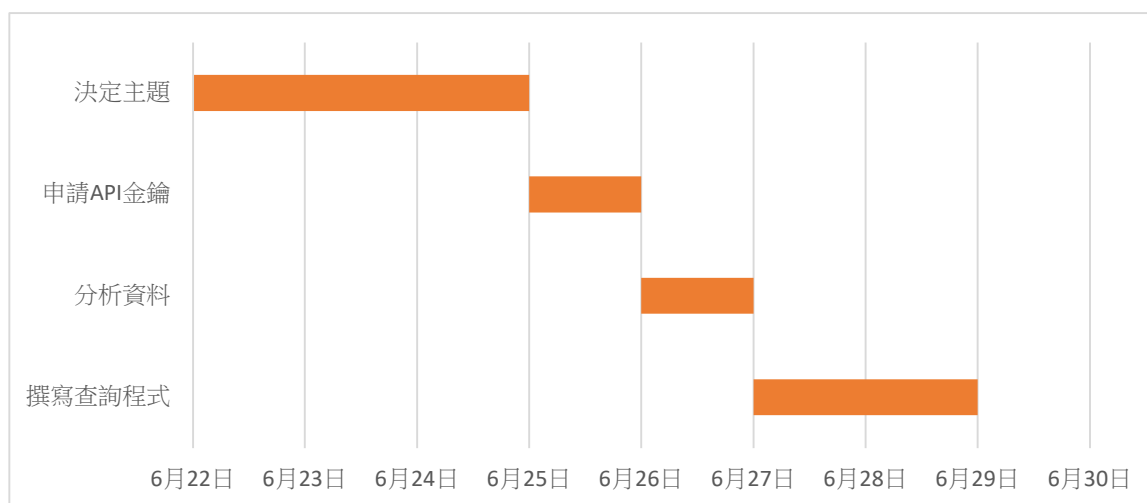


圖 1-2 計畫時程甘特圖

壹.第八節 研究範疇與限制

公共運輸整合流通資訊平臺所提供的 API 金鑰是有限制的，一般會員每日上限能夠呼叫 20,000 次，若該程式提供給更多使用者，必須將網路爬蟲取得的資料定時更新到自行建立的資料庫中，程式再從資料庫中取得資料。

第貳章、系統設計與架構

貳.第一節 系統架構

1. 蒐集：

利用網路爬蟲技術，利用平臺所提供之 API 取得各個
所需要的資料。

2. 統整：

將取得的大量資料整理成該程式所需用到的資訊。

3. 製作：

將資訊運用到程式中。

第參章、 系統實作與展示

參.第一節 平台的授權設定

藉由公共運輸整合資訊流通服務平臺所提供的範例程式以及註冊會員所申請到的 app_id 和 app_key 來使用該平臺的資料。

```
app_id = '5827be7a93e04190a20003d4dc00a26f'
app_key = 'ShHsxdH-X9b9Jfz_rUgh1-vRwvK8'

class Auth():
    def __init__(self, app_id, app_key):
        self.app_id = app_id
        self.app_key = app_key

    def get_auth_header(self):
        xdate = format_date_time(mktime(datetime.now().timetuple()))
        hashed = hmac.new(self.app_key.encode('utf8'), ('x-date: ' + xdate).encode('utf8'), sha1)
        signature = base64.b64encode(hashed.digest()).decode()

        authorization = 'hmac username="' + self.app_id + '", ' + \
            'algorithm="hmac-sha1", ' + \
            'headers="x-date", ' + \
            'signature="' + signature + "'"

        return {
            'Authorization': authorization,
            'x-date': format_date_time(mktime(datetime.now().timetuple())),
            'Accept - Encoding': 'gzip'
        }
```

圖 3-1、平台授權程式碼

參.第二節 主要程式展示

以同時使用多個 API 獲取車站、車次、車次抵達時間、車次誤點狀況等資訊。


```

while True:
    i_stationname = input("請輸入車站名稱(例如：田中): ")
    if __name__ == '__main__':
        a = Auth(app_id, app_key)

        # 取得指定車站資訊
        response = request('get', 'https://ptx.transportdata.tw/MOTC/v2/Rail/TRA/Station?format=JSON', headers=a.get_auth_header())
        if response.status_code == 200:
            data = json.loads(response.content)
            station_name = dict(zip([i['StationName'] for i in data], data))

        # 取得指定車站即將到站車次
        response_station = request('get', f'https://ptx.transportdata.tw/MOTC/v2/Rail/TRA/LiveBoard/Station/{station_name[i_stationname]['StationID']}?format=JSON', headers=a.get_auth_header())
        if response_station.status_code == 200:
            data = json.loads(response_station.content)

            # 取得車次編號清單(當前位置)
            response_train = request('get', f'https://ptx.transportdata.tw/MOTC/v2/Rail/TRA/LiveTrainDelay?format=JSON', headers=a.get_auth_header())
            if response_train.status_code == 200:
                data_train = json.loads(response_train.content)
                train = dict(zip([i['TrainNo'] for i in data_train], data_train))
                # print(train)
            else:
                print("Train error")

            print(f">>> 即將抵達 {station_name[i_stationname]['StationName']} 的列車 <<<")
            for d in data:
                # 取得列車到站時間
                response_trainNo = request('get', f'https://ptx.transportdata.tw/MOTC/v2/Rail/TRA/Daily timetable/today/trainNo/{d['TrainNo']}?format=JSON', headers=a.get_auth_header())
                if response_trainNo.status_code == 200:
                    data_trainNo = json.loads(response_trainNo.content)
                    station_info = dict(zip([i['StationName'] for i in data_trainNo[0]['Stoptimes']], data_trainNo[0]['Stoptimes']))
                else:
                    print("Train error")

                direction = '北上' if d['Direction'] == 0 else '南下'
                print(f">>> {d['TrainNo']} - 方向: {direction} - 車種: {d['TrainTypeLine'] ['Zh_tw']} - 抵達時間: {station_info[i_stationname]['ArrivalTime']} - \
                # 起點: {train[d['TrainNo']] ['DelayTime']} 分鐘 - 終點站: {d['EndingStationName'] ['Zh_tw']} - 目前位置: {train[d['TrainNo']] ['StationName'] ['Zh_tw']}>>>")
                print()
            else:
                print(response_station.status_code)
        else:
            print(response_station.status_code)

```

圖 3-2、系統程式碼

參.第三節 系統執行結果

讓使用者輸入台灣鐵路車站名稱，以獲取即將到站車次

目前所在位置、到站時間、誤點狀況等等資訊。

```

請輸入車站名稱(例如：田中)：彰化
>>> 即將抵達 彰化 的列車 <<<
車次：2551，方向：南下，車種：區間，抵達時間：21:36，誤點：0 分鐘，終點站：彰化，目前位置：竹南>>
車次：181，方向：南下，車種：自強(推拉式自強號且無自行車車廂)，抵達時間：21:58，誤點：0 分鐘，終點站：屏東，目前位置：成功>>
車次：374，方向：北上，車種：自強(DMU3100 型柴聯)，抵達時間：22:02，誤點：0 分鐘，終點站：臺中，目前位置：花蓮>>
車次：283，方向：南下，車種：普悠瑪，抵達時間：22:12，誤點：0 分鐘，終點站：斗六，目前位置：臺中>>
車次：2553，方向：南下，車種：區間，抵達時間：22:17，誤點：2 分鐘，終點站：彰化，目前位置：龍井>>
車次：2257，方向：南下，車種：區間，抵達時間：22:20，誤點：0 分鐘，終點站：嘉義，目前位置：臺中>>
車次：2284，方向：北上，車種：區間，抵達時間：22:22，誤點：0 分鐘，終點站：新竹，目前位置：社頭>>

請輸入車站名稱(例如：田中)：中壢
>>> 即將抵達 中壢 的列車 <<<
車次：1261，方向：南下，車種：區間，抵達時間：21:33，誤點：2 分鐘，終點站：中壢，目前位置：內壢>>
車次：155，方向：南下，車種：自強(推拉式自強號且無自行車車廂)，抵達時間：21:39，誤點：1 分鐘，終點站：彰化，目前位置：新豐>>
車次：2254，方向：北上，車種：區間，抵達時間：22:08，誤點：0 分鐘，終點站：基隆，目前位置：楊梅>>
車次：1271，方向：南下，車種：區間，抵達時間：22:17，誤點：0 分鐘，終點站：北湖，目前位置：鶯歌>>
車次：1282，方向：北上，車種：區間，抵達時間：22:21，誤點：0 分鐘，終點站：七堵，目前位置：北湖>>
車次：1267，方向：南下，車種：區間，抵達時間：22:28，誤點：0 分鐘，終點站：苗栗，目前位置：板橋>>
車次：150，方向：北上，車種：自強(推拉式自強號且無自行車車廂)，抵達時間：22:31，誤點：2 分鐘，終點站：七堵，目前位置：香山>>

```

圖 3-3、系統執行結果

第肆章、 結論與未來展望

在執行這項期末專題初期，為了主題煩惱了許久，最後決定以公共運輸作為這次專題的大綱，因為期末較為繁忙，無法抽空做這次專題，於是沒有做太多細節，若未來有打算繼續開發這項系統，能往圖形化介面發展，並將系統改以其他程式語言開發為行動裝置應用程式。

第伍章、 參考文獻

1. Public Transport API 介接範例程式碼說明

<https://motc-ptx-api-documentation.gitbook.io/motc-ptx-api-documentation/api-shi-yong/code>

2. YOUBIKE 多維度戰情系統 - 大同大學

<http://www.mis.ttu.edu.tw/ezfiles/72/1072/img/664/146366388.pdf>