

112 學年 APP 程式設計 -- 期末書面報告

主題：台鐵時刻表查詢軟體

組員：數四甲 S0822022 黃振翔 數四甲 S0822026 王正楷

數四甲 S0822031 詹士賢




一、製作動機	2
二、功能介紹	2
三、程式製作流程	3
四、可改進的地方	6
五、分工表	6
六、結語	7

一、製作動機：

因為平常很常搭乘火車往返彰化與家鄉，所以想要試著寫寫看像台鐵 e 訂通一樣的 APP，可以透過這個程式查詢到各站點間的火車班次，與某站點某段時間的時刻表。

二、功能介紹：

1. 車次查詢：	2. 車次查詢：	3. 車次查詢：
輸入列車車次後，會跳出該台列車之表定時間。	輸入列車車次後，會跳出該台列車之表定時間。	輸入列車車次後，會跳出該台列車之表定時間。
		
4. 紀錄車次：	在車次查詢中點選任意班次即可將該車次記錄下來，點選右下角綠色按鈕後即可看到其中儲存的车次	

三、程式製作流程：

1. 取得 TDX 運輸資料流通服務平臺之 ACCESS TOKEN

為了取得這個平台的 ACCESS TOKEN 我們必須先註冊帳號並申請到金鑰，才能無上限的使用這個 API（如右圖）。並且我們在程式的中加入了這段 function 使這個程式可以使用我們的帳號與金鑰來自動取得 API 的使用權（如下圖）。

Available authorizations

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. API requires the following scopes. Select which ones you want to grant to Swagger UI.

TDX (OAuth2, clientCredentials)

TDX會員可至【會員專區-資料服務-服務金鑰】功能頁面，從預設金鑰(或建立新的金鑰)取得Client Id和Client Secret資訊，分別輸入至下方client_id和client_secret欄位。按下Authorize按鈕，即可開始於Swagger介面呼叫TDX API。

Token URL: <https://tdx.transportdata.tw/auth/realms/TDXConnect/protocol/openid-connect/token>

Flow: clientCredentials

client_id:

.....

client_secret:

.....

Authorize

Close

```
fun AccessToken(callback: (accessToken: String?) -> Unit){  
    val client = OkHttpClient()  
    val url = "https://tdx.transportdata.tw/auth/realms/TDXConnect/protocol/openid-connect/token"  
    val requestBody = FormBody.Builder()  
        .add( name: "grant_type", value: "client_credentials")  
        .add( name: "client_id", value: ".....")  
        .add( name: "client_secret", value: ".....")  
        .build()  
    val request = Request.Builder()  
        .url(url)  
        .addHeader( name: "content-type", value: "application/x-www-form-urlencoded")  
        .post(requestBody)  
        .build()  
}
```

TDX 平臺網址: <https://tdx.transportdata.tw/>

2. 放進 URL

這段程式碼的目的是為了讓三個 fragment 都能抓到 url 的 access token。

```
AccessToken { accessToken ->  
    this@MainActivity.accessToken = accessToken  
    val firstFragment = FirstFragment.newInstance( param1: "param1", param2: "param2")  
    val secondFragment = SecondFragment.newInstance( param1: "param1", param2: "param2")  
    val thirdFragment = ThirdFragment.newInstance( param1: "param1", param2: "param2")  
    supportFragmentManager.beginTransaction()  
        .replace(R.id.activity_main_nav_fragment, thirdFragment)  
        .add(R.id.activity_main_nav_fragment, secondFragment)  
        .add(R.id.activity_main_nav_fragment, firstFragment)  
        .commit()  
}
```

這段則是將 main activity 抓到 access token 並放到 fragment 之 url 內。

```
val trainTimeTableList: MutableList<String> = mutableListOf()
val accessToken = (activity as? MainActivity)?.accessToken
val client = OkHttpClient()
val url =
    "https://tdx.transportdata.tw/api/basic/v3/Rail/TRA/GeneralTrainTimetable/TrainNo/${stationID}?%24format=JSON"
val request = Request.Builder()
    .url(url)
    .addHeader(name: "accept", value: "application/json")
    .addHeader(
        name: "Authorization",
        value: "Bearer $accessToken"
    )
    .build()
```

3. 製作車次查詢功能

- I. 放 editText
- II. 把 editText 限定數字鍵盤

- III. 輸入後會跳出 listView

串接 API 服務說明位址：

(已展開的項目即為該 API)

<https://tdx.transportdata.tw/api-service/swagger/basic/5fa88b0c-120b-43f1-b188->

[c379ddb2593d#/TRA/GeneralTrainTimetableApiController_Get_3208_1](https://tdx.transportdata.tw/api-service/swagger/basic/5fa88b0c-120b-43f1-b188-c379ddb2593d#/TRA/GeneralTrainTimetableApiController_Get_3208_1)



4. 製作站間車次功能

- I. 放三個 textView 並監聽他們的點擊
- II. 點選前兩個 textView 會跳出清單
- III. 點選第三個會跳出選擇日期
- IV. 建立並監聽確定按鈕
- V. 點擊確定按鈕後會關掉其他元件
甲、並跳出 listView

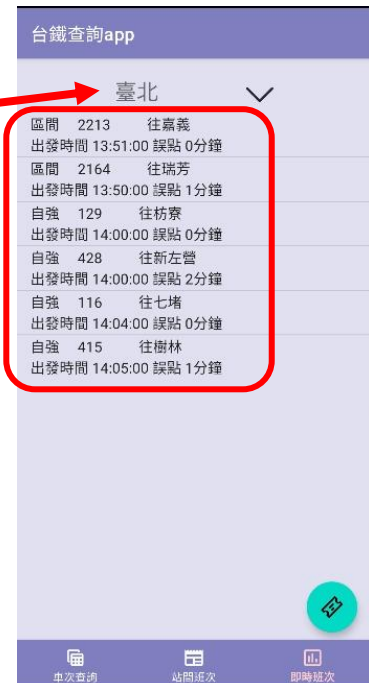


串接 API 服務說明位址： (已展開的項目即為該 API)

https://tdx.transportdata.tw/api-service/swagger/basic/5fa88b0c-120b-43f1-b188-c379ddb2593d#/TRA/DailyTrainTimeTableApiController_Get_3211_5

5. 製作即時班次功能

- I. 放 textView 並監聽點擊
- II. 點擊 textView 後會跳出車站選單
- III. 選擇後會跳出 listView

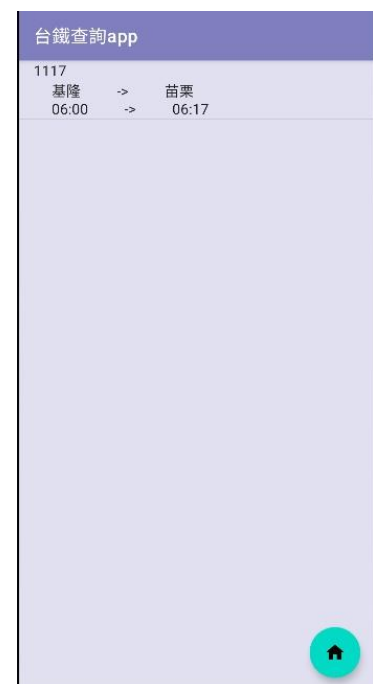


串接 API 服務說明位址： (已展開的項目即為該 API)

https://tdx.transportdata.tw/api-service/swagger/basic/5fa88b0c-120b-43f1-b188-c379ddb2593d#/TRA/StationLiveBoardApiController_Get_3213_1

6. 製作儲存車次功能

- I. 建立 textView 並顯示它
- II. 建立 floatingActionButton
做跳回主頁面的功能
- III. 點選 listView 中的選項後可以刪除該筆資料



四、可改進的地方：

1. 車站選擇

在挑選起點車站與終點車站時，因為我們沒有做篩選縣市的功能，所以會在選擇時跑出全台灣的車站，這方面使用上相當不便利。

2. 會員登入功能

我們並沒有做會員登入系統，應該要透過 firebase 做出登入系統來讓各自的帳戶儲存各自的資料(儲存車班)。

3. 版面美化

背景的部分有點單調並且留白太多，可以再更進一步調整版面配置，還有下面的 bar 選單該放大一點並把圖示放大，不然現在的版本字有點太小看不太清楚。

4. 車輛動態

可以在儲存車班那邊再做一個監聽該列車的功能，可以在儲存車班的畫面顯示他現在在哪一站，並且預計多久以後會到我們要上車的站點。

五、分工表：

數四甲	黃振翔	S0822022	程式架構、串接 API
數四甲	王正楷	S0822026	製作簡報、書面報告
數四甲	詹士賢	S0822031	程式美觀、書面報告

六、 結語：

這是我們三個第一次做出完整的 program，所以我們在這次報告中都在實作方面獲得了很大的進步，了解了製作一個 app 的流程，但最後做出來的程式仍然還有許多的不足能做改善，之後應該會再另外找時間把這個程式寫的完善一點，最後謝謝老師這學期的指導，我們三個在這學期的課程中受益良多，還學習到了如何對抗新冠肺炎。