113年智慧行動商務系統開發班第1期

行動裝置APP開發與製作學習評量

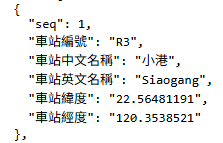
|  |  |  |  |
| --- | --- | --- | --- |
| **評量編號** | 2-7-2 | | |
| **評量主題** | Android APP設計及實作應用 | **實施時間** | 2025/8/25 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/8/25 |
| **評量目的** | 評定學員是否能開發一個Android APP，從題目所指定的Web API獲取資料(JSON或XML等格式)，並將這些資料使用適當的UI元件進行資料的排版，呈現在APP的介面上。 | | |
| **評量說明** | **任務一：**分析題目指定的Web API服務所回傳的資料，依照其格式將資料解析(parse)成後續任務實作所需的資料單元序列，以利後續程式設計之用。  **任務二：**使用非同步請求從題目指定的Web API服務取得資料，且需處理API連線過程中的可能出現錯誤情況，提示相關訊息。  **任務三：**使用適當的介面元件顯示取得的資料於APP介面上，並進行資料單元排版美化以及單元點擊事件的處理。  **加分任務：**其他配套製作，額外實現功能。 | | |
| **提交要求** | 繳交時請以FTP上傳**相關程式檔案**與**作業文件檔**各一份，**相關程式檔案**請另放置在「完成結果檔」資料夾內，最外層資料夾名稱請用**兩碼學號與姓名**，例如1號朱OO，資料夾請命名為「01黃OO」。 | | |

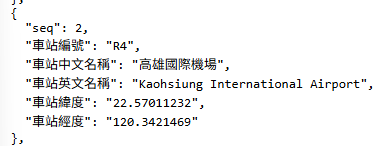
座號：24 姓名：鍾惟安

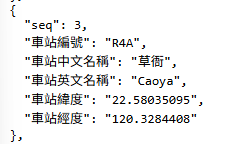
**任務一：分析題目指定的Web API服務所回傳的資料，依照其格式將資料解析(parse)成後續任務實作所需的資料單元序列，以利後續程式設計之用。需達到下列各項要求：**

1. 連線高雄城市資料平台的Web API服務查詢高雄紅橘線捷運車站中心座標。
2. Web API服務網址 : https://api.kcg.gov.tw/api/service/Get/4278fc6a-c3ea-4192-8ce0-40f00cdb40dd
3. 將回傳的捷運車站資料進行解析(parse)，取出後續任務所需的資料。資料區塊需包含"車站編號"、"車站中文名稱、"車站緯度"、"車站經度"等內容，再將資料區塊整理成Json單元序列，最後將單元序列前10筆貼在任務一的答題程式碼區塊。

結果畫面截圖（至少三張圖）：







程式碼：

[

{

"seq": 1,

"車站編號": "R3",

"車站中文名稱": "小港",

"車站英文名稱": "Siaogang",

"車站緯度": "22.56481191",

"車站經度": "120.3538521"

},

{

"seq": 2,

"車站編號": "R4",

"車站中文名稱": "高雄國際機場",

"車站英文名稱": "Kaohsiung International Airport",

"車站緯度": "22.57011232",

"車站經度": "120.3421469"

},

{

"seq": 3,

"車站編號": "R4A",

"車站中文名稱": "草衙",

"車站英文名稱": "Caoya",

"車站緯度": "22.58035095",

"車站經度": "120.3284408"

},

{

"seq": 4,

"車站編號": "R5",

"車站中文名稱": "前鎮高中",

"車站英文名稱": "Cianjhen Senior High School",

"車站緯度": "22.58853833",

"車站經度": "120.3219713"

},

{

"seq": 5,

"車站編號": "R6",

"車站中文名稱": "凱旋",

"車站英文名稱": "Kaisyuan",

"車站緯度": "22.59683914",

"車站經度": "120.3151478"

},

{

"seq": 6,

"車站編號": "R7",

"車站中文名稱": "獅甲",

"車站英文名稱": "Shihjia",

"車站緯度": "22.60583276",

"車站經度": "120.307702"

},

{

"seq": 7,

"車站編號": "R8",

"車站中文名稱": "三多商圈",

"車站英文名稱": "Sanduo Shopping District",

"車站緯度": "22.61383541",

"車站經度": "120.3046764"

},

{

"seq": 8,

"車站編號": "R9",

"車站中文名稱": "中央公園",

"車站英文名稱": "Central Park",

"車站緯度": "22.6245709",

"車站經度": "120.3006424"

},

{

"seq": 9,

"車站編號": "R10",

"車站中文名稱": "美麗島",

"車站英文名稱": "Formosa Boulevard",

"車站緯度": "22.631386",

"車站經度": "120.301951"

},

{

"seq": 10,

"車站編號": "R11",

"車站中文名稱": "高雄車站",

"車站英文名稱": "Kaohsiung Main Station",

"車站緯度": "22.63966255",

"車站經度": "120.3027023"

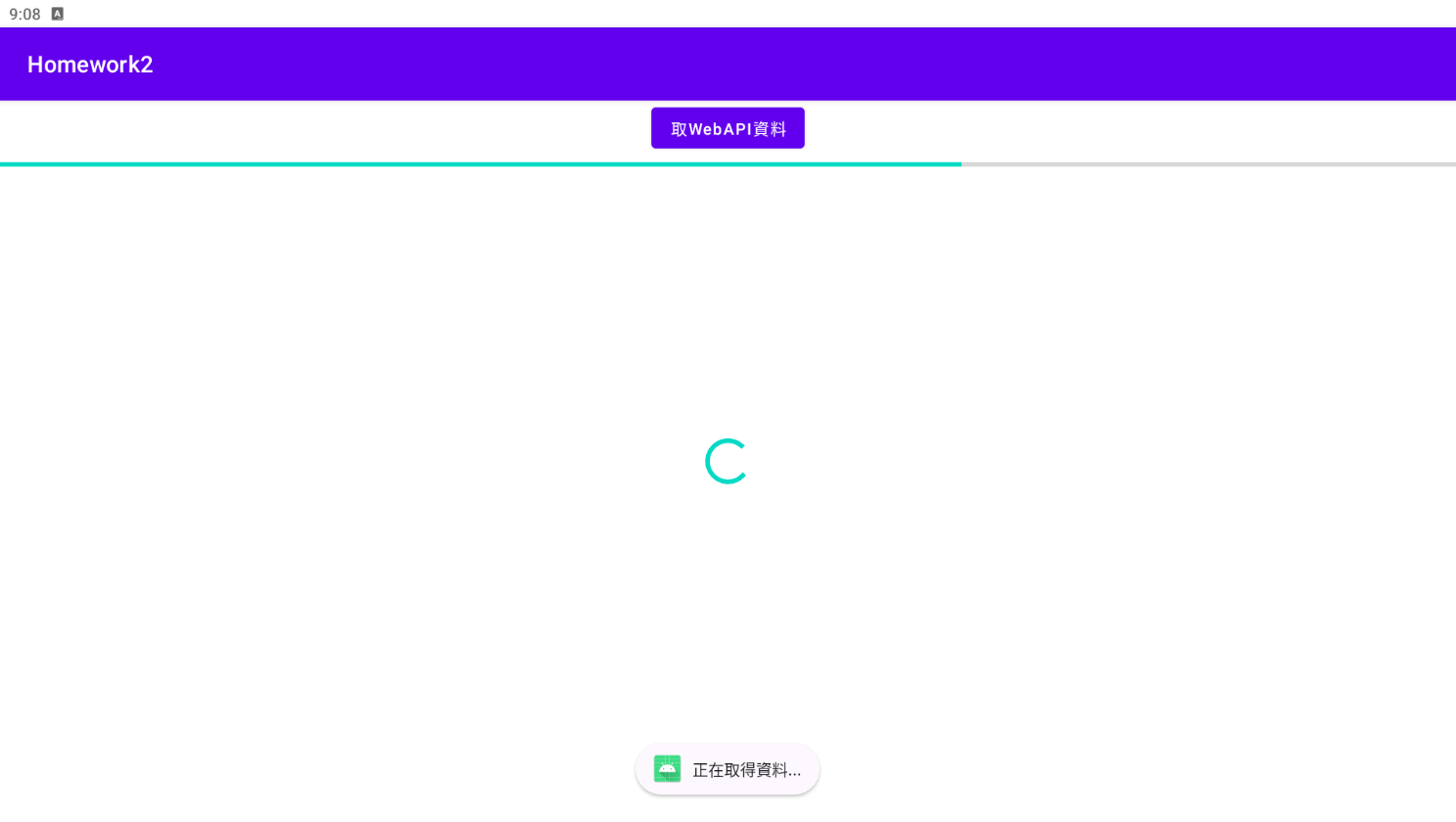
}

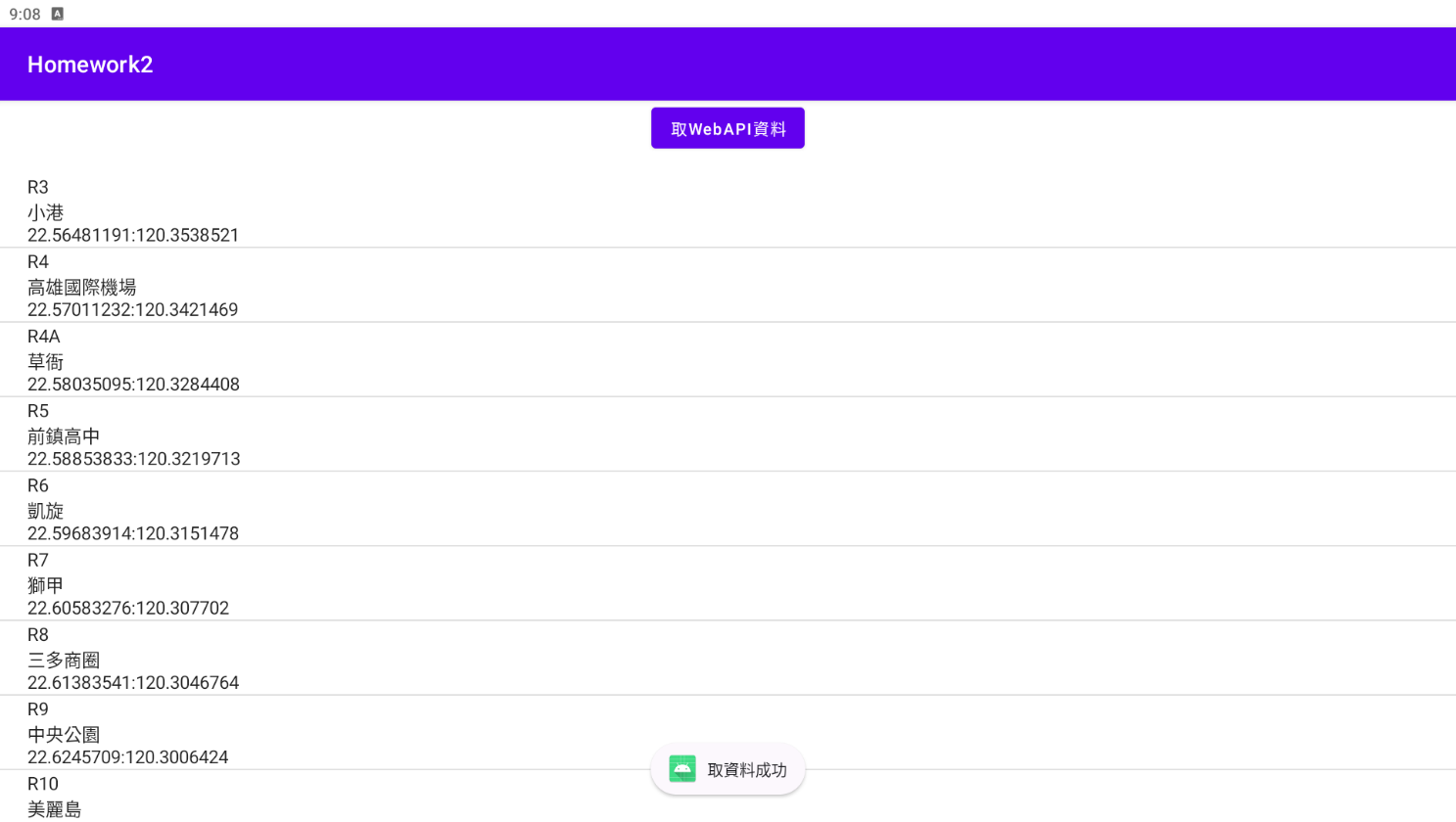
]

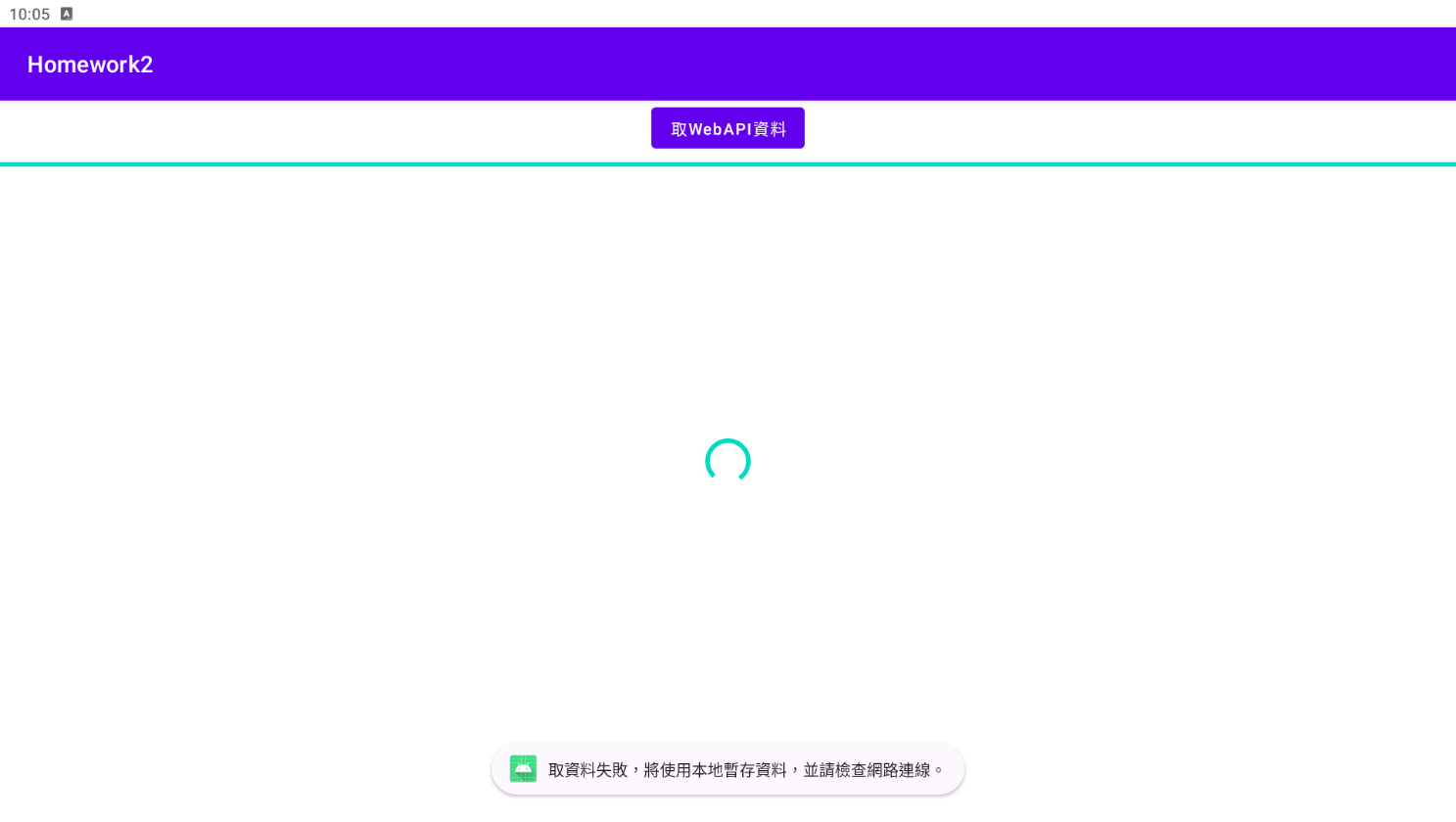
**任務二：使用非同步請求從題目指定的Web API服務取得資料，且需處理API連線過程中的可能出現錯誤情況，提示相關訊息。需達到下列各項要求：**

1. 利用Java撰寫程式進行API服務介接並取回捷運車站中心座標Json資料。
2. 撰寫方法將回傳的捷運車站資料依照任務一的說明進行解析(parse)，並整理成任務三使用RecyclerView或ListView顯示時所需的資料集合。
3. 必須處理API介接時任何可能的錯誤及例外情況，例如無法連接到API，JSON格式錯誤、回應狀況處理等。
4. 當發生錯誤時，顯示適當的錯誤訊息。

結果畫面截圖（至少三張圖）：







程式碼：

//取WebAPI資料

public void GetData(View View) {

Toast.makeText(this, "正在取得資料...", Toast.LENGTH\_SHORT).show();

ShowProgressBar(true, 0f);

//清空已顯示的資料

DataList.setAdapter(null);

new Thread(() -> {

final String[] ToastMessage = new String[1];

//變數在 try 區塊和 catch 區塊都可能被賦值，Java 不允許在 lambda 外部這樣重複賦值。

//建議改用陣列或將 ToastMessage 宣告在 try-catch 外並只賦值一次。

try {

URL URL = new URL(API\_URL);

HttpURLConnection Connection = (HttpURLConnection) URL.openConnection();

Connection.setRequestMethod("GET");

Connection.setConnectTimeout(5000); // 設定連線超時時間為 5 秒

Connection.setReadTimeout(10000); // 設定讀取超時時間為 10 秒

Connection.connect();

int ResponseCode = Connection.getResponseCode();

if (ResponseCode == HttpURLConnection.HTTP\_OK) {

int Total = Connection.getContentLength();

BufferedReader Reader = new BufferedReader(new InputStreamReader(Connection.getInputStream()));

StringBuffer Buffer = LoadString(Reader, Total);

if (Buffer.length() == 0) {

ToastMessage[0] = "取資料失敗";

} else {

ToastMessage[0] = "取資料成功";

ShowData(Buffer.toString());

}

//SwipeRefresh.setRefreshing(false); // 停止刷新動畫

} else {

Log.e(TAG, "取資料失敗 Code : " + ResponseCode);

ToastMessage[0] = LoadLocalData();

}

if (Connection != null) {

Log.v(TAG, "斷開連線");

Connection.disconnect();

}

} catch (UnknownHostException E) {

Log.e(TAG, "網路錯誤 : " + E.toString());

ToastMessage[0] = LoadLocalData();

//Sleep(1000);//模擬延遲

} catch (Exception E) {

Log.e(TAG, "取資料失敗 : " + E.toString());

ToastMessage[0] = LoadLocalData();

}

runOnUiThread(() -> {

ShowProgressBar(false, null);

Toast.makeText(this, ToastMessage[0], Toast.LENGTH\_SHORT).show();

});

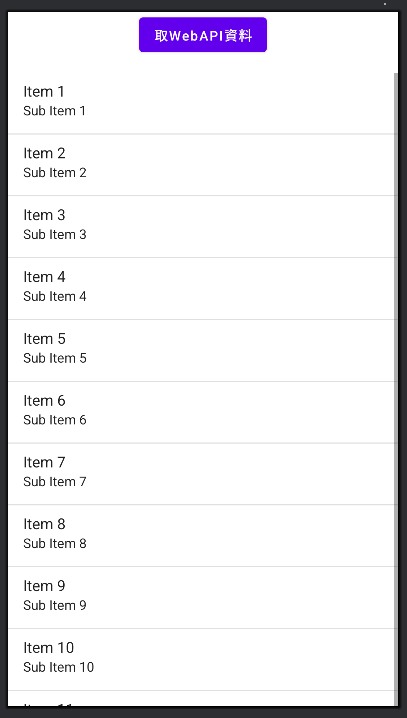
}).start();

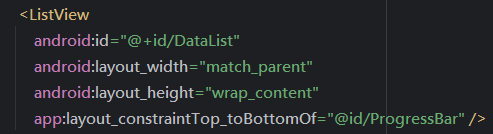
}

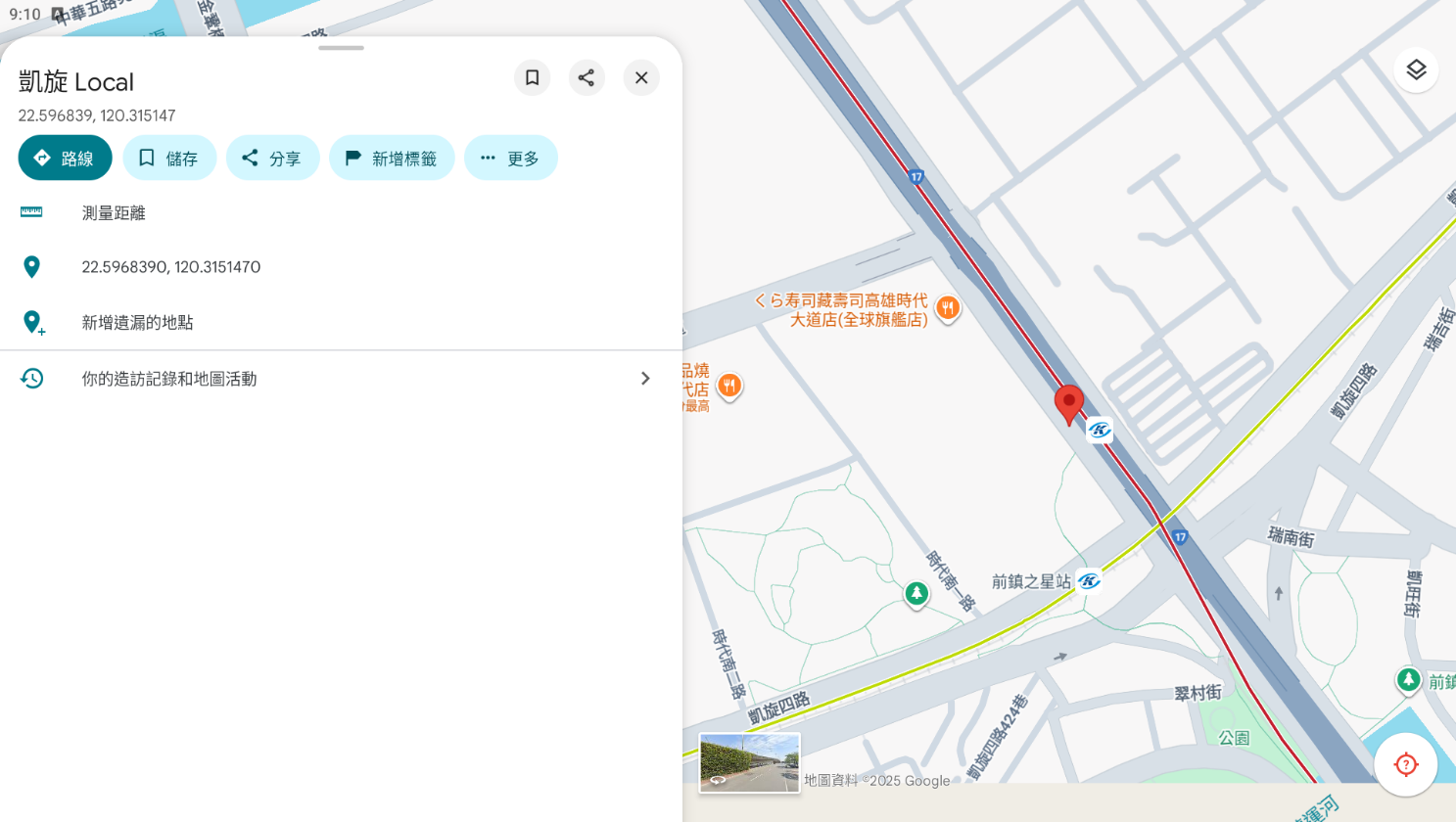
**任務三：使用適當的介面元件顯示取得的資料於APP介面上，並進行資料單元排版美化以及單元點擊事件的處理。需達到下列各項要求**

1. 使用先前所學的排版方式(RecyclerView或ListView排版元件)，適當地呈現獲取到的資料。
2. 畫面編排樣式不限，但需合乎一般使用習慣與原則。
3. 若資料正在傳輸中須以適當方式告知使用者，如顯示正在下載中等字眼或圖示。
4. 排版顯示的捷運站列表項目必須可以被點擊，點擊後在新的Activity畫面顯示該項目的詳細地圖位置。

結果畫面截圖（至少三張圖）：







程式碼：

<ListView

android:id="@+id/DataList"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/ProgressBar" />

void ShowMap(String Title, double Lat, double Lng) {

try {

var GeoString = "geo:" + Lat + "," + Lng;

var QueryString = Lat + "," + Lng + "(" + Title + ")";

var UriString = Uri.encode(QueryString);

GeoString += "?q=" + UriString;

//開啟地圖

startActivity(new Intent(Intent.ACTION\_VIEW, Uri.parse(GeoString)));

} catch (Exception E) {

Log.e(TAG, "ShowMapError : " + E.getMessage());

Toast.makeText(this, "顯示地圖失敗", Toast.LENGTH\_SHORT).show();

}

}

<ProgressBar

android:id="@+id/ProgressBar"

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:max="100"

android:progress="0"

android:visibility="invisible"

app:layout\_constraintTop\_toBottomOf="@id/Btn\_GetData" />

<ProgressBar

android:id="@+id/Loading"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:visibility="invisible"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

//進度條開關, 顯示進度

void ShowProgressBar(Boolean OnOff, Float Value) {

//物件開關

ProgressBar.setVisibility(OnOff ? View.VISIBLE : View.INVISIBLE);

//讀取圈

Loading.setVisibility(OnOff ? View.VISIBLE : View.INVISIBLE);

if (Value != null) {

Log.v(TAG, "SetProgress : " + Value);

ProgressBar.setProgress(Value.intValue(), true);

}

}

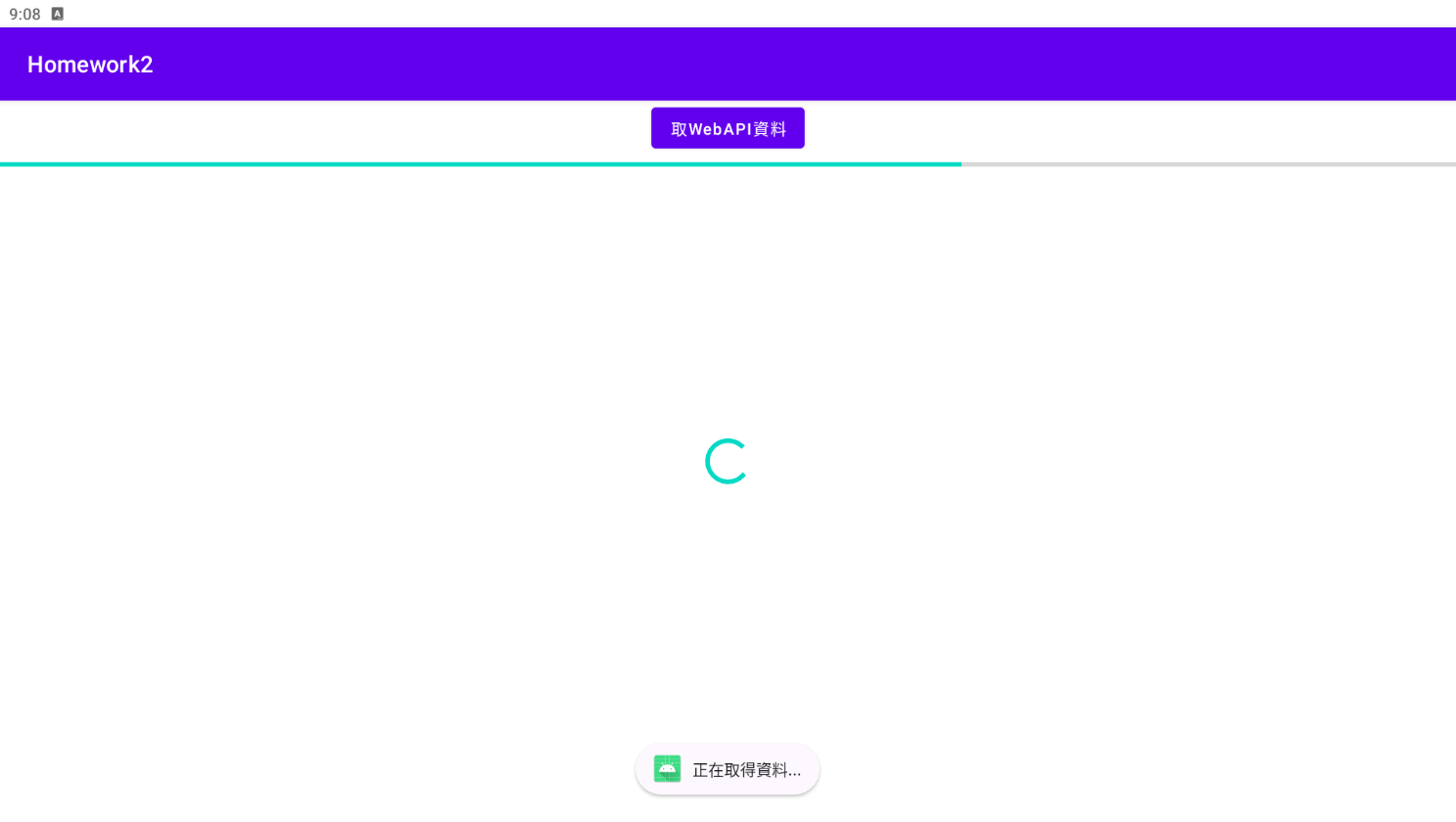
**加分任務：其他配套製作，額外實現功能。達到下列各項要求**

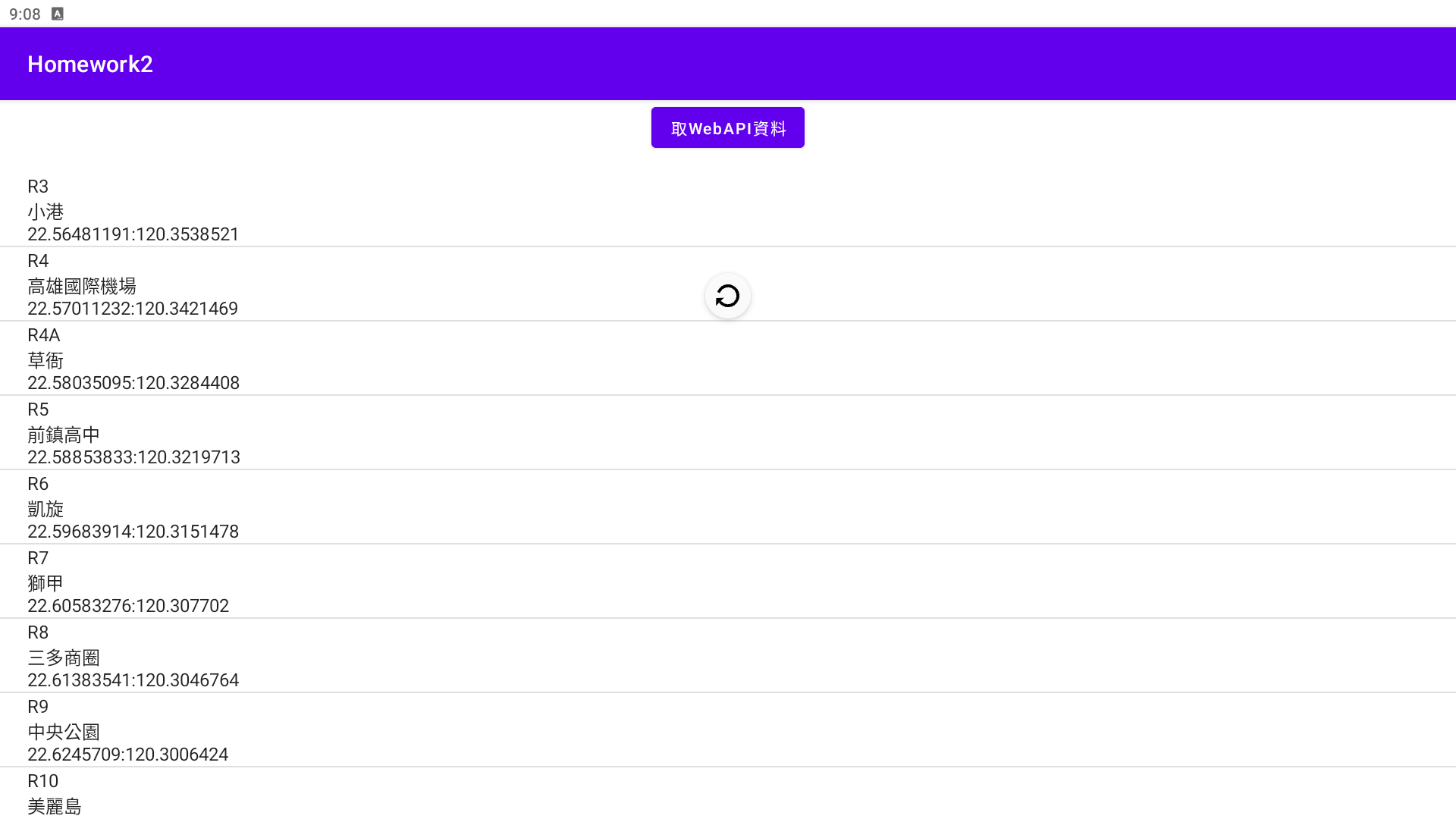
1. 在資料加載過程中，顯示一個ProgressBar進度指示器。
2. 資料加載完成後，隱藏ProgressBar進度指示器並顯示資料。
3. 提供SwipeRefreshLayout的資料刷新功能。
4. 支援離線模式，在沒有網絡連接時顯示Local端的暫存資料。

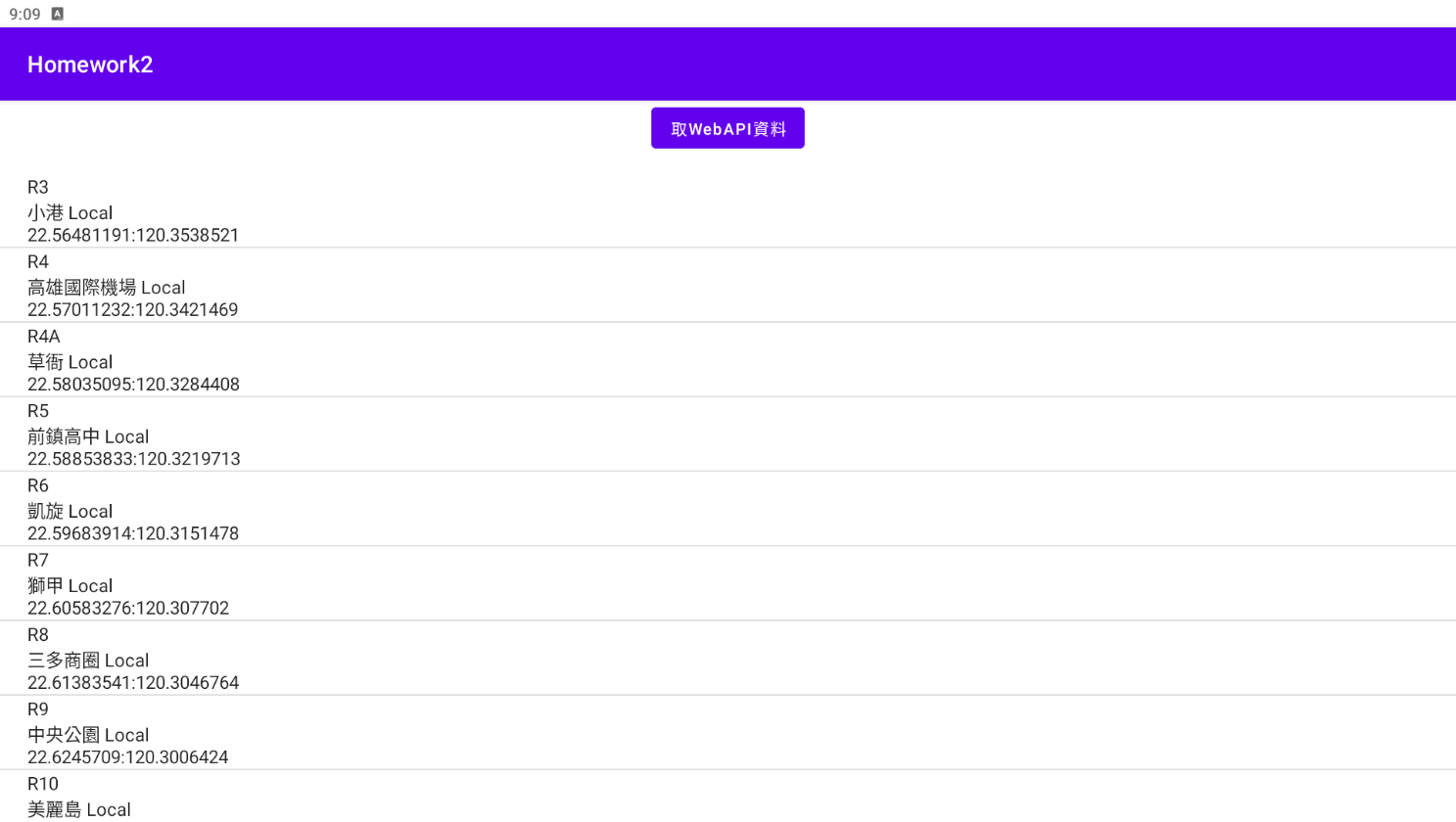
**介面參考圖**

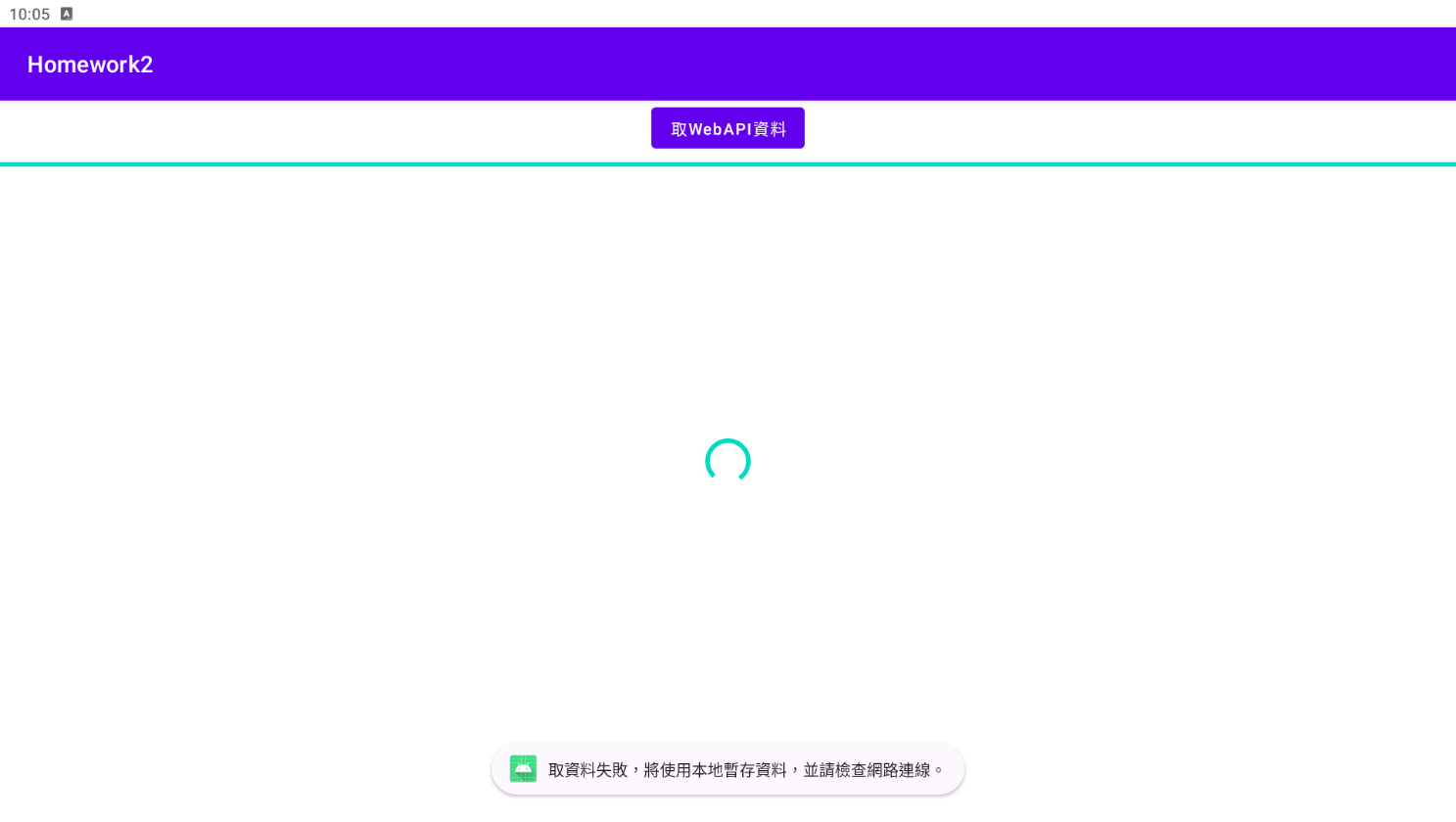


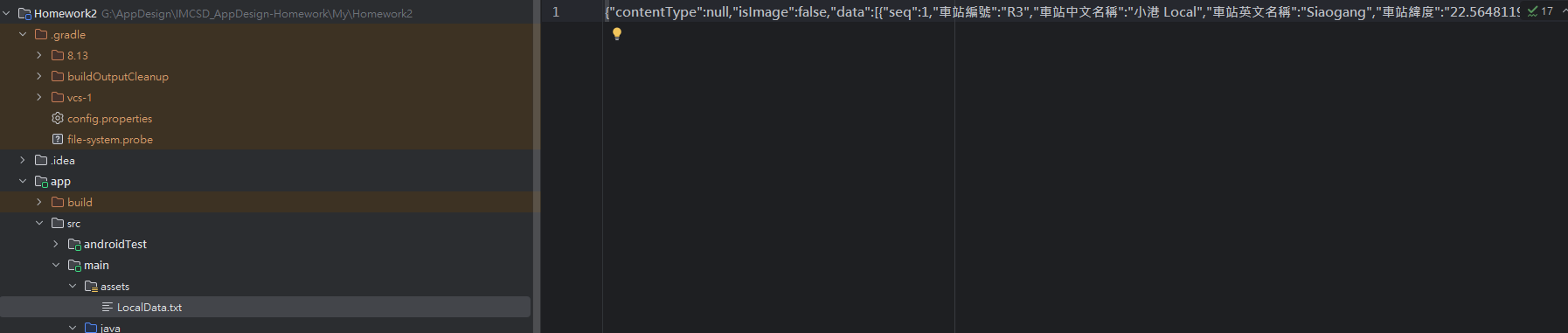
結果畫面截圖（至少三張圖）：











程式碼：

SwipeRefresh = findViewById(R.id.SwipeRefresh);

SwipeRefresh.setOnRefreshListener(() -> {

Log.d(TAG, "下滑更新");

GetData(null);

SwipeRefresh.setRefreshing(false); // 停止刷新動畫

});

String LoadLocalData() {

Sleep(1000);//模擬延遲

runOnUiThread(() -> Toast.makeText(this, "取資料失敗，將使用本地暫存資料，並請檢查網路連線。", Toast.LENGTH\_SHORT).show());

Sleep(1000);//模擬延遲

try {

InputStream IS = getAssets().open("LocalData.txt");

BufferedReader Reader = new BufferedReader(new InputStreamReader(IS, "UTF-8"));

//本地讀取用StringBuilder即可，但為求統一方法所以使用StringBuffer

StringBuffer Buffer = LoadString(Reader, 0);

Sleep(1000);

if (Buffer.length() == 0) {

return "讀取本地暫存資料失敗";

} else {

ShowData(Buffer.toString());

return "讀取本地暫存資料完成";

}

} catch (Exception E) {

Sleep(1500);//模擬延遲

Log.e(TAG, "LoadLocalData E : " + E.toString());

return "讀取本地暫存資料失敗";

}

}

<androidx.swiperefreshlayout.widget.SwipeRefreshLayout

android:id="@+id/SwipeRefresh"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/ProgressBar">

<ListView

android:id="@+id/DataList"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/ProgressBar" />

</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>