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

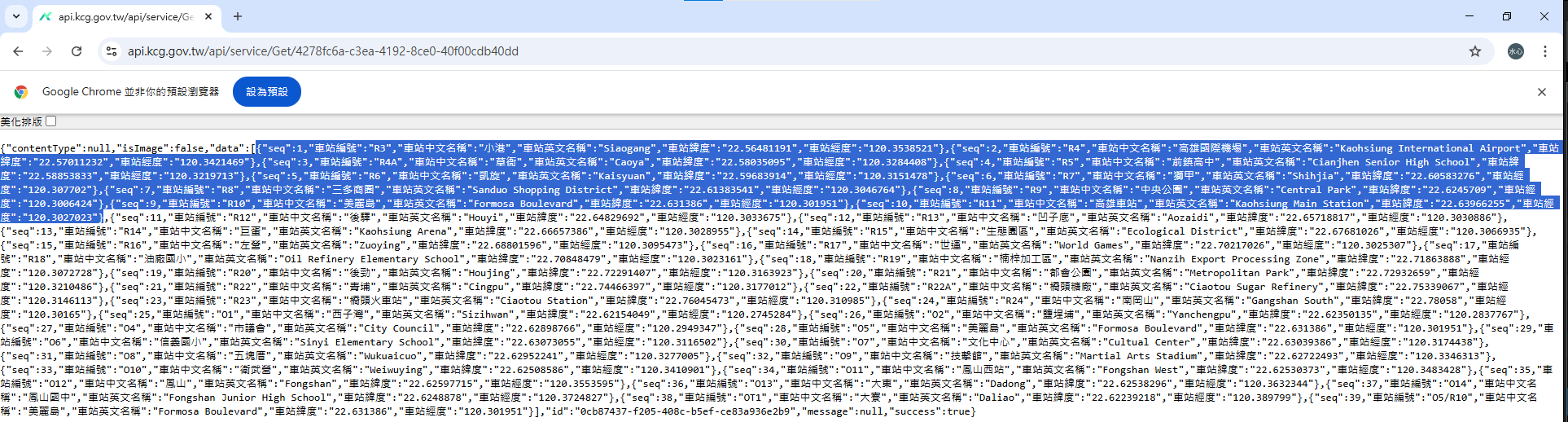
|  |  |  |  |
| --- | --- | --- | --- |
| **評量編號** | 2-7-2 | | |
| **評量主題** | Android APP設計及實作應用 | **實施時間** | 2025/2/17 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/2/17 |
| **評量目的** | 評定學員是否能開發一個Android APP，從題目所指定的Web API獲取資料(JSON或XML等格式)，並將這些資料使用適當的UI元件進行資料的排版，呈現在APP的介面上。 | | |
| **評量說明** | **任務一：**分析題目指定的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： 15 姓名： 陳延銓

**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"}

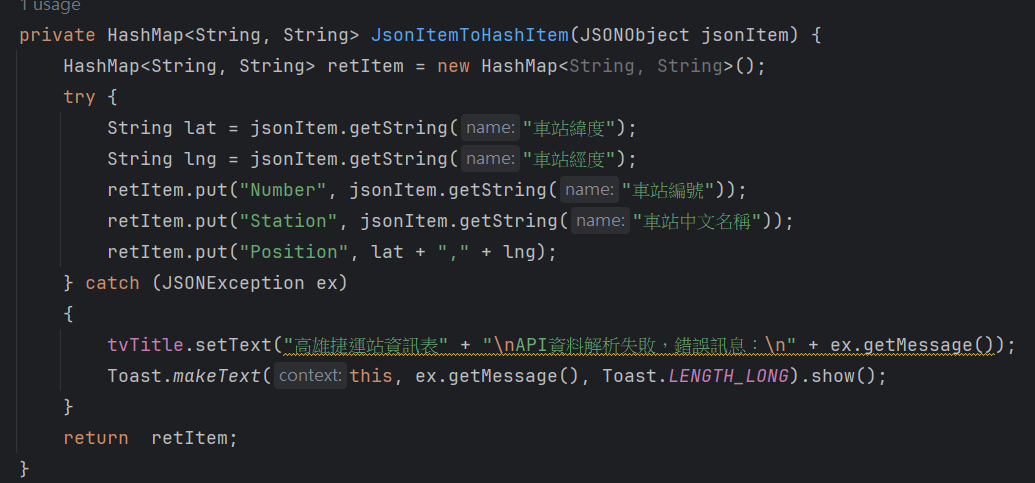
**任務二：使用非同步請求從題目指定的Web API服務取得資料，且需處理API連線過程中的可能出現錯誤情況，提示相關訊息。需達到下列各項要求：**

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：

private String ReadBackUPPosition() {

String json = "";

try (FileInputStream fis = openFileInput(fileName);

BufferedReader reader = new BufferedReader(new InputStreamReader(fis))) {

StringBuilder sb = new StringBuilder();

String line = "";

while ((line = reader.readLine()) != null) {

sb.append(line);

}

json = sb.toString();

} catch (FileNotFoundException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI離線資料解析失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

} catch (IOException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI離線資料讀取失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

}

private HashMap<String, String> JsonItemToHashItem(JSONObject jsonItem) {

HashMap<String, String> retItem = new HashMap<String, String>();

try {

String lat = jsonItem.getString("車站緯度");

String lng = jsonItem.getString("車站經度");

retItem.put("Number", jsonItem.getString("車站編號"));

retItem.put("Station", jsonItem.getString("車站中文名稱"));

retItem.put("Position", lat + "," + lng);

} catch (JSONException ex)

{

tvTitle.setText("高雄捷運站資訊表" + "\nAPI資料解析失敗，錯誤訊息：\n" + ex.getMessage());

Toast.makeText(this, ex.getMessage(), Toast.LENGTH\_LONG).show();

}

return retItem;

}

private boolean HandleAdapter(JSONObject response)

{

try {

JSONArray arr = response.getJSONArray("data");

for (int i = 0; i < arr.length(); i++)

{

JSONObject jsonItem = arr.getJSONObject(i);

HashMap<String, String> tmpItem = JsonItemToHashItem(jsonItem);

listStationInfo.add(tmpItem);

}

adapterInfo.notifyDataSetChanged();

tvTitle.setText("高雄捷運站資訊表\n資料讀取完畢");

return true;

} catch (JSONException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI舊資料解析失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

}

}

private void JsonRequestLink() {

try {

JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, strUrl, null, new Response.Listener<JSONObject>() {

@Override

public void onResponse(JSONObject response) {

if (!HandleAdapter(response)) {

String json = ReadBackUPPosition();

try {

JSONObject object = new JSONObject(json);

HandleAdapter(object);

tvTitle.setText("高雄捷運站資訊表\n離線資料讀取完畢");

} catch (JSONException e) {

throw new RuntimeException(e);

}

} else {

WriteJsonToFile(response.toString());

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI資料取得失敗，錯誤訊息：\n" + error.getMessage());

String json = ReadBackUPPosition();

try {

JSONObject object = new JSONObject(json);

HandleAdapter(object);

tvTitle.setText("高雄捷運站資訊表\n離線資料讀取完畢");

} catch (JSONException e) {

throw new RuntimeException(e);

}

}

});

Volley.newRequestQueue(this).add(request);

tvTitle.setText("高雄捷運站資訊表" + "\nAPI資料讀取中....");

} catch (Exception ex) {

tvTitle.setText("高雄捷運站資訊表" + "\n嘗試發送API請求失敗，錯誤訊息：\n" + ex.getMessage());

String json = ReadBackUPPosition();

try {

JSONObject object = new JSONObject(json);

HandleAdapter(object);

} catch (JSONException e) {

throw new RuntimeException(e);

}

}

}

**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：

MainAcitvity.java:

package com.example.examapplication;

import android.content.Context;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.os.SystemClock;

import android.view.View;

import android.widget.Button;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

import com.android.volley.Request;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.JsonObjectRequest;

import com.android.volley.toolbox.Volley;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.HashMap;

public class MainActivity extends AppCompatActivity {

final String fileName = "MRKPosition.txt";

TextView tvTitle;

RecyclerView recyclerView;

SwipeRefreshLayout swipeRefreshLayout;

ProgressBar progressBar;

InfoAdapter adapterInfo;

String strUrl = "https://api.kcg.gov.tw/api/service/Get/4278fc6a-c3ea-4192-8ce0-40f00cdb40dd";

ArrayList<HashMap<String, String>> listStationInfo;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// EdgeToEdge.enable(this);

setContentView(R.layout.activity\_main);

// ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {

// Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());

// v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);

// return insets;

// });

listStationInfo = new ArrayList<HashMap<String, String>>();

tvTitle = findViewById(R.id.tv\_title);

recyclerView = findViewById(R.id.recycle\_view);

progressBar = findViewById(R.id.progressBar);

swipeRefreshLayout = findViewById(R.id.sr\_layout);

swipeRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

@Override

public void onRefresh() {

progressBar.setVisibility(View.VISIBLE);

new Thread(new Runnable() {

@Override

public void run() {

SystemClock.sleep(1000);

progressBar.setVisibility(View.INVISIBLE);

}

}).start();

JsonRequestLink();

swipeRefreshLayout.setRefreshing(false);

}

});

adapterInfo = new InfoAdapter(this);

adapterInfo.SetListData(listStationInfo);

adapterInfo.setOnItemListener(position -> {

String strItem = listStationInfo.get(position).get("Position");

String strStation = listStationInfo.get(position).get("Station");

String strLabel = strItem + "(" + strStation + ")";

String strQuery = Uri.decode(strLabel);

Intent i = new Intent(Intent.ACTION\_VIEW, Uri.parse("geo:" + strItem + "?q=" + strQuery + "&z=32"));

startActivity(i);

});

recyclerView.setLayoutManager(new LinearLayoutManager(this));

recyclerView.setAdapter(adapterInfo);

Button btnGetInfo = findViewById(R.id.btn\_get\_data);

btnGetInfo.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

listStationInfo.clear();

progressBar.setVisibility(View.VISIBLE);

new Thread(new Runnable() {

@Override

public void run() {

SystemClock.sleep(1000);

progressBar.setVisibility(View.INVISIBLE);

}

}).start();

JsonRequestLink();

progressBar.setVisibility(View.INVISIBLE);

}

});

Button btnClearData = findViewById(R.id.btn\_clear\_data);

btnClearData.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

listStationInfo.clear();

adapterInfo.notifyDataSetChanged();

tvTitle.setText("高雄捷運站資訊表\n資料已清空");

}

});

}

private void JsonRequestLink() {

try {

JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, strUrl, null, new Response.Listener<JSONObject>() {

@Override

public void onResponse(JSONObject response) {

if (!HandleAdapter(response)) {

String json = ReadBackUPPosition();

try {

JSONObject object = new JSONObject(json);

HandleAdapter(object);

tvTitle.setText("高雄捷運站資訊表\n離線資料讀取完畢");

} catch (JSONException e) {

throw new RuntimeException(e);

}

} else {

WriteJsonToFile(response.toString());

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI資料取得失敗，錯誤訊息：\n" + error.getMessage());

String json = ReadBackUPPosition();

try {

JSONObject object = new JSONObject(json);

HandleAdapter(object);

tvTitle.setText("高雄捷運站資訊表\n離線資料讀取完畢");

} catch (JSONException e) {

throw new RuntimeException(e);

}

}

});

Volley.newRequestQueue(this).add(request);

tvTitle.setText("高雄捷運站資訊表" + "\nAPI資料讀取中....");

} catch (Exception ex) {

tvTitle.setText("高雄捷運站資訊表" + "\n嘗試發送API請求失敗，錯誤訊息：\n" + ex.getMessage());

String json = ReadBackUPPosition();

try {

JSONObject object = new JSONObject(json);

HandleAdapter(object);

} catch (JSONException e) {

throw new RuntimeException(e);

}

}

}

private boolean HandleAdapter(JSONObject response)

{

try {

JSONArray arr = response.getJSONArray("data");

for (int i = 0; i < arr.length(); i++)

{

JSONObject jsonItem = arr.getJSONObject(i);

HashMap<String, String> tmpItem = JsonItemToHashItem(jsonItem);

listStationInfo.add(tmpItem);

}

adapterInfo.notifyDataSetChanged();

tvTitle.setText("高雄捷運站資訊表\n資料讀取完畢");

return true;

} catch (JSONException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI舊資料解析失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

}

}

private HashMap<String, String> JsonItemToHashItem(JSONObject jsonItem) {

HashMap<String, String> retItem = new HashMap<String, String>();

try {

String lat = jsonItem.getString("車站緯度");

String lng = jsonItem.getString("車站經度");

retItem.put("Number", jsonItem.getString("車站編號"));

retItem.put("Station", jsonItem.getString("車站中文名稱"));

retItem.put("Position", lat + "," + lng);

} catch (JSONException ex)

{

tvTitle.setText("高雄捷運站資訊表" + "\nAPI資料解析失敗，錯誤訊息：\n" + ex.getMessage());

Toast.makeText(this, ex.getMessage(), Toast.LENGTH\_LONG).show();

}

return retItem;

}

private String ReadBackUPPosition() {

String json = "";

try (FileInputStream fis = openFileInput(fileName);

BufferedReader reader = new BufferedReader(new InputStreamReader(fis))) {

StringBuilder sb = new StringBuilder();

String line = "";

while ((line = reader.readLine()) != null) {

sb.append(line);

}

json = sb.toString();

} catch (FileNotFoundException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI離線資料解析失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

} catch (IOException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI離線資料讀取失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

}

return json;

}

private boolean WriteJsonToFile(String json) {

try (FileOutputStream fos = openFileOutput(fileName, Context.MODE\_PRIVATE)) {

fos.write(json.getBytes());

} catch (IOException e) {

return false;

}

return true;

}

}

InfoAdapter.java

package com.example.examapplication;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

import java.util.HashMap;

public class InfoAdapter extends RecyclerView.Adapter<InfoItem> {

ArrayList<HashMap<String, String>> arrayList;

Context context;

private OnItemClickListener listener;

public interface OnItemClickListener {

void onItemClick(int position);

}

public void setOnItemListener(OnItemClickListener Listener) {

this.listener = Listener;

}

public InfoAdapter(Context c) {

context = c;

}

public void SetListData(ArrayList<HashMap<String, String>> list) {

arrayList = list;

notifyDataSetChanged();

}

@Override

public InfoItem onCreateViewHolder(ViewGroup parent, int viewType) {

View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.recycleview\_item\_layout, parent, false);

return new InfoItem(v, listener);

}

@Override

public void onBindViewHolder(InfoItem holder, int position) {

holder.tvNum.setText("車站編號：" + arrayList.get(position).get("Number"));

holder.tvPosition.setText("車站位置：" + arrayList.get(position).get("Position"));

holder.tvStation.setText("車站站名：" + arrayList.get(position).get("Station"));

}

@Override

public int getItemCount() {

return arrayList.size();

}

}

InfoItem.java

package com.example.examapplication;

import android.view.View;

import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

public class InfoItem extends RecyclerView.ViewHolder {

TextView tvNum, tvStation, tvPosition;

View v;

public InfoItem(View itemView, final InfoAdapter.OnItemClickListener listener) {

super(itemView);

v = itemView;

tvNum = itemView.findViewById(R.id.tv\_num);

tvStation = itemView.findViewById(R.id.tv\_station);

tvPosition = itemView.findViewById(R.id.tv\_lat\_lng);

itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (listener != null) {

int pos = getBindingAdapterPosition();

if (pos != RecyclerView.NO\_POSITION) {

listener.onItemClick(pos);

}

}

}

});

}

}

**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：

recyclerView = findViewById(R.id.recycle\_view);

progressBar = findViewById(R.id.progressBar);

swipeRefreshLayout = findViewById(R.id.sr\_layout);

swipeRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {

@Override

public void onRefresh() {

listStationInfo.clear();

progressBar.setVisibility(View.VISIBLE);

new Thread(new Runnable() {

@Override

public void run() {

SystemClock.sleep(1000);

progressBar.setVisibility(View.INVISIBLE);

}

}).start();

JsonRequestLink();

swipeRefreshLayout.setRefreshing(false);

}

});

adapterInfo = new InfoAdapter(this);

adapterInfo.SetListData(listStationInfo);

adapterInfo.setOnItemListener(position -> {

String strItem = listStationInfo.get(position).get("Position");

String strStation = listStationInfo.get(position).get("Station");

String strLabel = strItem + "(" + strStation + ")";

String strQuery = Uri.decode(strLabel);

Intent i = new Intent(Intent.ACTION\_VIEW, Uri.parse("geo:" + strItem + "?q=" + strQuery + "&z=32"));

startActivity(i);

});

recyclerView.setLayoutManager(new LinearLayoutManager(this));

recyclerView.setAdapter(adapterInfo);

btnGetInfo.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

listStationInfo.clear();

progressBar.setVisibility(View.VISIBLE);

new Thread(new Runnable() {

@Override

public void run() {

SystemClock.sleep(1000);

progressBar.setVisibility(View.INVISIBLE);

}

}).start();

JsonRequestLink();

progressBar.setVisibility(View.INVISIBLE);

}

});

private String ReadBackUPPosition() {

String json = "";

try (FileInputStream fis = openFileInput(fileName);

BufferedReader reader = new BufferedReader(new InputStreamReader(fis))) {

StringBuilder sb = new StringBuilder();

String line = "";

while ((line = reader.readLine()) != null) {

sb.append(line);

}

json = sb.toString();

} catch (FileNotFoundException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI離線資料解析失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

} catch (IOException e) {

tvTitle.setText("高雄捷運站資訊表" + "\nAPI離線資料讀取失敗，錯誤訊息：\n" + e.getMessage());

throw new RuntimeException(e);

}

return json;

}

private boolean WriteJsonToFile(String json) {

try (FileOutputStream fos = openFileOutput(fileName, Context.MODE\_PRIVATE)) {

fos.write(json.getBytes());

} catch (IOException e) {

return false;

}

return true;

}

adapterInfo = new InfoAdapter(this);

adapterInfo.SetListData(listStationInfo);

adapterInfo.setOnItemListener(position -> {

String strItem = listStationInfo.get(position).get("Position");

String strStation = listStationInfo.get(position).get("Station");

String strLabel = strItem + "(" + strStation + ")";

String strQuery = Uri.decode(strLabel);

Intent i = new Intent(Intent.ACTION\_VIEW, Uri.parse("geo:" + strItem + "?q=" + strQuery + "&z=32"));

startActivity(i);

});

recyclerView.setLayoutManager(new LinearLayoutManager(this));

recyclerView.setAdapter(adapterInfo);