

天氣APP (具GPS定位)



🔍 Keelung



23/05/2022 12:04

Day 26.9° Night 21.9°

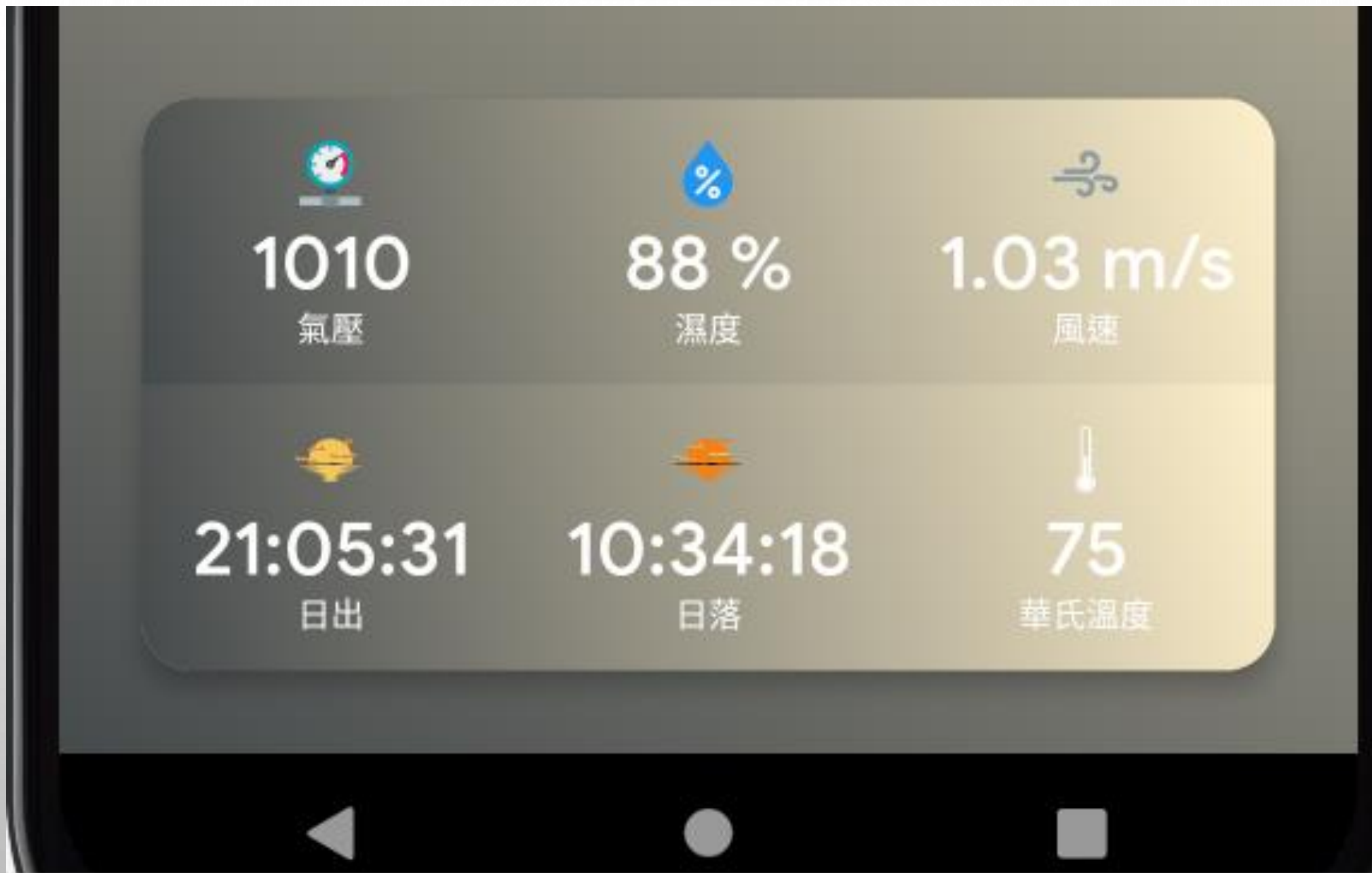
24.0°



體感溫度 24.7°

Clouds

- 可利用搜索框搜索地名
- 右上角的小地圖為定位按鈕
- 依天氣些切換顯示天氣圖片
(第12頁)



- 包含6項天氣屬性

利用 OPENWEATHERMAP

- 經緯度:

HTTPS://API.OPENWEATHERMAP.ORG/DATA/2.5/WEATHER?LAT={LAT}&LON={LON}&APPID={API KEY}

HTTPS://API.OPENWEATHERMAP.ORG/DATA/2.5/WEATHER?LAT=25.0478&LON=121.5319&APPID=8118ED6EE68DB2DEBFAAA5A44C832918

- 城市名:

HTTPS://API.OPENWEATHERMAP.ORG/DATA/2.5/WEATHER?Q={CITY NAME}&APPID={API KEY}

HTTPS://API.OPENWEATHERMAP.ORG/DATA/2.5/WEATHER?Q=TAIPEI&UNITS=METRIC&APPID=8118ED6EE68DB2DEBFAAA5A44C832918

```
395 companion object{
396     private const val PERMISSION_REQUEST_ACCESS_LOCATION = 100
397     const val API_KEY = "8118ed6ee68db2debfaaa5a44c832918"
398 }
```

https://api.openweathermap.org/data/2.5/weather?lat=25.0478&lon=121.5319&appid=8118ed6ee68db2debfaaa5a44c832918

```
{
  "coord": {
    "lon": 121.5324,
    "lat": 25.0504
  },
  "weather": [
    {
      "id": 803,
      "main": "Clouds",
      "description": "broken clouds",
      "icon": "04n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 296.64,
    "feels_like": 297,
    "temp_min": 294.85,
    "temp_max": 298.73,
    "pressure": 1009,
    "humidity": 75
  },
  "visibility": 10000,
  "wind": {
    "speed": 0.89,
    "deg": 170,
    "gust": 1.79
  },
  "clouds": {
    "all": 75
  },
  "dt": 1653308115,
  "sys": {
    "type": 2,
    "id": 266033,
    "country": "TW",
    "sunrise": 1653253589,
    "sunset": 1653302099,
    "timezone": 28800,
    "id": 1668341,
    "name": "Taipei",
    "cod": 200
  }
}
```

JSON formatter

JSON BEAUTIFIER

JSON PARSER

XML FORMATTER

JSBEAUTIFIER

SAVE

RECENT LINKS

LOGIN

Sample

```
1 {"coord":{"lon":121.5324,"lat":25.0504},"weather":
  :[{"id":803,"main":"Clouds","description":"broken
clouds","icon":"04n"}],"base":"stations","main"
:{"temp":296.64,"feels_like":297,"temp_min":294.85
,"temp_max":298.73,"pressure":1009,"humidity":75}
,"visibility":10000,"wind":{"speed":0.89,"deg":170
,"gust":1.79},"clouds":{"all":75},"dt":1653308115
,"sys":{"type":2,"id":266033,"country":"TW","sunrise"
:1653253589,"sunset":1653302099},"timezone":28800
,"id":1668341,"name":"Taipei","cod":200}
```

Upload Data

Validate

2 Tab Space

Format / Beautify

Minify / Compact

Convert JSON to

Download

JSON Full Form

Code

```
1 {
2   "coord": {
3     "lon": 121.5324,
4     "lat": 25.0504
5   },
6   "weather": [
7     {
8       "id": 803,
9       "main": "Clouds",
10      "description": "broken clouds",
11      "icon": "04n"
12    }
13  ],
14  "base": "stations",
15  "main": {
16    "temp": 296.64,
17    "feels_like": 297,
18    "temp_min": 294.85,
19    "temp_max": 298.73,
20    "pressure": 1009,
21    "humidity": 75
22  },
23  "visibility": 10000,
24  "wind": {
25    "speed": 0.89
```

ApiUtilities.kt x ApiInterface.kt x ModelClass.kt x

AndroidManifest.xml
com.example.weatherkotlin
POJO
Main
ModelClass
Sys
Weather
Wind
Utilities
ApiInterface
ApiUtilities

1

2

3

4

5

6

7

8

9

10

11

12

```
package com.example.weatherkotlin.POJO

import com.google.gson.annotations.SerializedName

data class ModelClass (
    @SerializedName("weather") val weather:List<Wear
    @SerializedName("main") val main:Main,
    @SerializedName("wind") val wind:Wind,
    @SerializedName("sys") val sys:Sys,
    @SerializedName("id") val id:Int,
    @SerializedName("name") val name:String
)
```


Android

app

- manifests
 - AndroidManifest.xml
- java
 - com.example.weatherkotlin
 - POJO
 - Main
 - ModelClass
 - Sys
 - Weather
 - Wind
 - Utilities
 - ApiInterface
 - ApiUtilities
 - MainActivity
- com.example.weatherkotlin (androidT)
- com.example.weatherkotlin (test)
- java (generated)
- res
 - res (generated)
- Gradle Scripts

Main.kt

```
1 package com.example.weatherkotlin.POJO
2
3 import com.google.gson.annotations.SerializedName
4
5 data class Main (
6     @SerializedName("temp") val temp:Double,
7     @SerializedName("feels_like") val feels_like:Double,
8     @SerializedName("temp_min") val temp_min:Double,
9     @SerializedName("temp_max") val temp_max:Double,
10    @SerializedName("pressure") val pressure:Int,
11    @SerializedName("humidity") val humidity:Int
12)
```

Weather.kt

```
1 com.example.weatherkotlin.POJO
2
3 com.google.gson.annotations.SerializedName
4
5 class Weather(
6     rializedName("id") val id:Int,
7     rializedName("main") val main:String,
8     rializedName("description") val description:String,
9     rializedName("icon") val icon:String
10)
```

Sys.kt

```
1 package com.example.weatherkotlin.POJO
2
3 import com.google.gson.annotations.SerializedName
4
5 data class Sys (
6     @SerializedName("type") val type:Int,
7     @SerializedName("id") val id:Int,
8     @SerializedName("country") val country:String,
9     @SerializedName("sunrise") val sunrise:Int,
10    @SerializedName("sunset") val sunset:Int
11)
```

Wind.kt

```
1 package com.example.weatherkotlin.POJO
2
3 import com.google.gson.annotations.SerializedName
4
5 data class Wind (
6     @SerializedName("speed") val speed:Double,
7     @SerializedName("deg") val deg:Int
8)
```


定位，取得“新位置”（非系統暫存定位位置）

```
145 @SuppressLint("MissingPermission")
146 private fun getLocation() {
147     //Log.d("times", "getLocation")
148     val locationRequestSettings = LocationRequest.create().apply { this: LocationRequest
149         fastestInterval = 1000
150         interval = 1000
151         priority = LocationRequest.PRIORITY_HIGH_ACCURACY
152         smallestDisplacement = 1.0f
153         numUpdates = 2 //取得位置2次(以免沒取到值)
154     }
155
156     fusedLocationProviderClient.requestLocationUpdates(
157         locationRequestSettings, locationCallback, Looper.myLooper()!!
158     )
159 }
160 private val locationCallback = object: LocationCallback() {
161     override fun onLocationResult(p0: LocationResult) {
162         var lastLocation: Location = p0.lastLocation
163         //Log.d("times", "locationCallback")
164         fetchCurrentLocationWeather(lastLocation.latitude.toString(), lastLocation.longitude.toString())
165     }
166 }
```

```

159
160 private val locationCallback = object: LocationCallback(){
161     override fun onLocationResult(p0: LocationResult) {
162         var lastLocation:Location = p0.lastLocation
163         //Log.d("times","locationCallback")
164         fetchCurrentLocationWeather(lastLocation.latitude.toString(),lastLocation.longitude.toString())
165     }
166 }

167
168
169 private fun fetchCurrentLocationWeather(latitude: String, longitude: String) {
170
171     activityMainBinding.pbLoading.visibility=View.VISIBLE
172     ApiUtilities.getApiInterface()?.getCurrentWeatherData(latitude, longitude, API_KEY)?.enqueue(object:
173         Callback<ModelClass> {
174             @RequiresApi(Build.VERSION_CODES.O)
175             override fun onResponse(call: Call<ModelClass>, response: Response<ModelClass>) {
176                 if(response.isSuccessful){
177                     setDataOnViews(response.body())
178                 }
179             }
180
181             override fun onFailure(call: Call<ModelClass>, t: Throwable) {
182                 Toast.makeText(applicationContext, text: "ERROR", Toast.LENGTH_SHORT).show()
183             }
184
185         })
186 }

```

```

1 package com.example.weatherko
2
3 import ...
4
5
6
7
8 interface ApiInterface {
9
10     @GET("weather")
11     fun getCurrentWeatherData(
12         @Query("lat") latitude:String,
13         @Query("lon") longitude:String,
14         @Query("APPID") api_key:String
15     ):Call<ModelClass>
16
17
18     @GET("weather")
19     fun getCityWeatherData(
20         @Query("q") cityName:String,
21         @Query("APPID") api_key:String
22     ):Call<ModelClass>
23
24 }

```

```
187
188 @RequiresApi(Build.VERSION_CODES.O)
189 private fun setDataOnViews(body: ModelClass?) {
190
191     val sdf = SimpleDateFormat( pattern: "dd/MM/yyyy hh:mm")
192     val currentDate = sdf.format(Date())
193     activityMainBinding.tvDateAndTime.text=currentDate
194
195     activityMainBinding.tvDayMaxTemp.text="Day "+kelvinToCelsius(body!!.main.temp_max)+"°"
196     activityMainBinding.tvDayMinTemp.text="Night "+kelvinToCelsius(body.main.temp_min)+"°"
197     activityMainBinding.tvTemp.text=" "+kelvinToCelsius(body.main.temp)+"°"
198     activityMainBinding.tvFeelsLike.text="體感溫度 "+kelvinToCelsius(body.main.feels_like)+"°"
199     activityMainBinding.tvWeatherType.text=body.weather[0].main
200     activityMainBinding.tvSunrise.text=timestampToLocalDate(body.sys.sunrise.toLong())
201     activityMainBinding.tvSunset.text=timestampToLocalDate(body.sys.sunset.toLong())
202     activityMainBinding.tvPressure.text=body.main.pressure.toString()
203     activityMainBinding.tvHumidity.text=body.main.humidity.toString()+" %"
204     activityMainBinding.tvWindSpeed.text=body.wind.speed.toString()+" m/s"
205
206     activityMainBinding.tvTempFarenhite.text=
207     " "+((kelvinToCelsius(body.main.temp)).times( other: 1.8).plus( other: 32).roundToInt())
208
209     activityMainBinding.etGetCityName.setText(body.name)
210
211     updateUI(body.weather[0].id)
212 }
213
```

```
214 private fun updateUI(id: Int) {
215     if(id in 200..232){
216         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS)
217         window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS)
218         window.statusBarColor = resources.getColor(R.color.thunderstorm)
219
220         activityMainBinding.rlToolbar.setBackgroundColor(resources.getColor(R.color.thunderstorm))
221         activityMainBinding.rlSubLayout.background = ContextCompat.getDrawable(
222             context: this@MainActivity,
223             R.drawable.thunderstrom_bg
224         )
225         activityMainBinding.llMainBgBelow.background = ContextCompat.getDrawable(
226             context: this@MainActivity,
227             R.drawable.thunderstrom_bg
228         )
229         activityMainBinding.llMainBgAbove.background = ContextCompat.getDrawable(
230             context: this@MainActivity,
231             R.drawable.thunderstrom_bg
232         )
233         activityMainBinding.ivWeatherBg.setImageResource(R.drawable.thunderstrom_bg)
234         activityMainBinding.ivWeatherIcon.setImageResource(R.drawable.thunderstrom)
235     }else if(id in 300..322){
236         window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS)
237         window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS)
238         window.statusBarColor = resources.getColor(R.color.drizzle)
239
240         activityMainBinding.rlToolbar.setBackgroundColor(resources.getColor(R.color.drizzle))
```


> java (generated)

▼ res

▼ drawable

clear.png

clear_bg.xml

clouds.png

clouds_bg.xml

drizzle.png

drizzle_bg.xml

humidity.png

ic_baseline_search_24.xml

ic_launcher_background.xml

ic_launcher_foreground.xml (v24)

location.png

mist.png

mist_bg.xml

pressure.png

rain.png

rainy_bg.xml

snow.png

snow_bg.xml

sunrise.png

sunset.png

temp.png

temperature.png

thunderstrom.png

thunderstrom_bg.xml

windspeed.png

openweathermap.org/weather-conditions

Weather

Weather in your city

Guide

API

Dashboard

Marketplace

Pricing

Maps

Our In

Group 2xx: Thunderstorm

| ID | Main | Description | Icon |
|-----|--------------|---------------------------------|------|
| 200 | Thunderstorm | thunderstorm with light rain | 11d |
| 201 | Thunderstorm | thunderstorm with rain | 11d |
| 202 | Thunderstorm | thunderstorm with heavy rain | 11d |
| 210 | Thunderstorm | light thunderstorm | 11d |
| 211 | Thunderstorm | thunderstorm | 11d |
| 212 | Thunderstorm | heavy thunderstorm | 11d |
| 221 | Thunderstorm | ragged thunderstorm | 11d |
| 230 | Thunderstorm | thunderstorm with light drizzle | 11d |
| 231 | Thunderstorm | thunderstorm with drizzle | 11d |
| 232 | Thunderstorm | thunderstorm with heavy drizzle | 11d |

Group 3xx: Drizzle

| ID | Main | Description | Icon |
|-----|---------|-------------------------|------|
| 300 | Drizzle | light intensity drizzle | 09d |

搜尋框部分(輸入地名)

```
59
60     activityMainBinding.etGetCityName.setOnEditorActionListener({ v, actionId, keyEvent ->
61         if(actionId == EditorInfo.IME_ACTION_SEARCH){
62             getCityWeather(activityMainBinding.etGetCityName.text.toString())
63             val view=this.currentFocus
64             if(view != null){
65                 val imm:InputMethodManager=getSystemService(INPUT_METHOD_SERVICE) as InputMethodManager
66                 imm.hideSoftInputFromWindow(view.windowToken, flags: 0)
67                 activityMainBinding.etGetCityName.clearFocus()
68             }
69             true ^lambda
70         }
71         else false ^lambda
72     })
73 }
74
75 private fun getCityWeather(cityName: String) {
76     activityMainBinding.pbLoading.visibility=View.INVISIBLE
```

搜尋框避免輸入錯誤閃退(例外處理)

```
74
75 private fun |getCityWeather(cityName: String) {
76     activityMainBinding.pbLoading.visibility=View.VISIBLE
77     ApiUtilities.getApiInterface()?.getCityWeatherData(cityName, API_KEY)?.enqueue(object :Callback<ModelClass>{
78
79         @RequiresApi(Build.VERSION_CODES.0)
80         override fun onResponse(call: Call<ModelClass>, response: Response<ModelClass>) {
81             //setDataOnViews(response.body()) //可能有 java.lang.nullpointerexception
82
83             try {
84                 setDataOnViews(response.body())
85             } catch (e: Exception) {
86                 Toast.makeText(applicationContext, text: "Not a Valid City Name",Toast.LENGTH_SHORT).show()
87             }
88         }
89
90         override fun onFailure(call: Call<ModelClass>, t: Throwable) {
91             Toast.makeText(applicationContext, text: "Not a Valid City Name",Toast.LENGTH_SHORT).show()
92         }
93     })
94 }
95
```


進行定位部分

```
97 private fun getCurrentLocation(){
98     if(checkPermissions()){ 確認安卓系統 有無給予APP定位權限
99
100         if(isLocationEnabled()){ 確認定位功能有無打開
101
102             if(ActivityCompat.checkSelfPermission(
103                 context: this,
104                 android.Manifest.permission.ACCESS_FINE_LOCATION
105             ) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
106                 context: this,
107                 android.Manifest.permission.ACCESS_COARSE_LOCATION
108             ) != PackageManager.PERMISSION_GRANTED
109         ){
110             requestPermission()
111             return
112         }
113
114         getLocation()
115
116         fusedLocationProviderClient.lastLocation.addOnCompleteListener(this){ task ->
117             val location: Location? = task.result
118
119             if(location==null){
120                 Toast.makeText(context: this, text: "Null location Recieved", Toast.LENGTH_SHORT).show()
121
122                 //getLocation()
123                 //fetchCurrentLocationWeather("25.0478", "121.5319") //直接指定經緯度為台北市
124             }
125         }
126     }
127 }
```

```
124         }else{
125             //Toast.makeText(this,"Get location Successfully",Toast.LENGTH_SHORT).show()
126
127             fetchCurrentLocationWeather(location.latitude.toString(),location.longitude.toString())
128         }
129     }
130
131     }else{
132         //跳出手機內建的打開定位選單
133         Toast.makeText(context: this, text: "Turn on location",Toast.LENGTH_SHORT).show()
134         val intent = Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS)
135         startActivity(intent)
136     }
137
138     }else{
139         //請求權限
140         requestPermission()
141     }
142 }
```

第10頁 傳入 經緯度

確認安卓系統 有無給予APP定位權限

```
400 private fun checkPermissions(): Boolean{
401     if(ActivityCompat.checkSelfPermission(
402         context: this,
403         android.Manifest.permission.ACCESS_COARSE_LOCATION
404     ) == PackageManager.PERMISSION_GRANTED || ActivityCompat.checkSelfPermission(
405         context: this,
406         android.Manifest.permission.ACCESS_FINE_LOCATION
407     ) == PackageManager.PERMISSION_GRANTED)
408     {
409         return true
410     }
411     return false
412 }
413
414
```

```
385 private fun requestPermission(){
386     ActivityCompat.requestPermissions(
387         activity: this, arrayOf(
388             android.Manifest.permission.ACCESS_COARSE_LOCATION,
389             android.Manifest.permission.ACCESS_FINE_LOCATION
390         ),
391         PERMISSION_REQUEST_ACCESS_LOCATION
392     )
393 }
```

確認定位功能有無打開

- 定位有2種 (網路4G定位 或 GPS定位)

```
378 private fun isLocationEnabled(): Boolean{  
379     val locationManager: LocationManager=getSystemService(Context.LOCATION_SERVICE) as LocationManager  
380  
381     return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) || locationManager.isProviderEnabled(  
382         LocationManager.NETWORK_PROVIDER)  
383 }
```