心得:之前對於API或是JSON的資訊比較不了解，概念比較模糊，但經過這一次的功課，這讓我不清楚的部分減少了很多，酸然還是不怎麼熟悉但總該是有基本的概念出來了。

JAVA:  
registerReceiver(**receiver**,**new** IntentFilter(**"MyMessage"**));  
findViewById(R.id.***btn\_query***).setOnClickListener(**new** View.OnClickListener() {  
@Override  
**public void** onClick(View v) {  
 Request req =**new** Request.Builder().url(**"https://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=55ec6d6e-dc5c-4268-a725-d04cc262172b"**).build();  
 **new** OkHttpClient().newCall(req).enqueue(**new** Callback() {  
 @Override  
 **public void** onFailure(@NotNull Call call, @NotNull IOException e) {  
 Log.*e*(**"查詢失敗"**,e.toString());  
 }  
 @Override  
 **public void** onResponse(@NotNull Call call, @NotNull Response response) **throws** IOException {  
  
 sendBroadcast(**new** Intent(**"MyMessage"**).putExtra(**"json"**,response.body().string()));  
 }  
 });  
}  
});

Kotlin:

registerReceiver(**receiver**, IntentFilter(**"MyMessage"**))  
findViewById<Button>(R.id.*btn\_query*).setOnClickListener(View.OnClickListener **{  
 val** req = Request.Builder()  
 .url(**"https://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=55ec6d6e-dc5c-4268-a725-d04cc262172b"**)  
 .build()  
 OkHttpClient().newCall(req).enqueue(**object** : Callback {  
 **override fun** onFailure(call: Call, e: IOException) {  
 Log.e(**"查詢失敗"**, e.toString())  
 }  
  
 @Throws(IOException::**class**)  
 **override fun** onResponse(call: Call, response: Response) {  
  
 sendBroadcast(Intent(**"MyMessage"**).putExtra(**"json"**, response.**body**!!.string()))  
 }  
 })  
**}**)

Github: <https://github.com/BLPM/HomeWork>