1. okHttp3中请求数据成功的onResponse（Call call,Response response）回调方法中,如果返回的数据类型为String字符串，（如平时开发中最常见的是返回Json字符串），通过response.body().string()方法获取到返回结果，但是

该方法只能调用一次，第二次调用会获取不到结果

所以通常的做法是将返回结果保存到一个成员变量中，下次直接引用成员变量即可，而不用再次调用body().string()方法

Request request = new Request.Builder().url(requestURL).build();  
Call call = client.newCall(request);  
call.enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 Message msg = Message.*obtain*();  
 msg.what = *REQUEST\_FAILURE\_CODE*;  
 uiHandler.sendMessage(msg);  
 }  
  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 String result = response.body().string()；

Message msg = Message.*obtain*();  
 msg.what = *REQUEST\_SUCCESS\_CODE*;  
 msg.obj = result ;

Log.e(TAG,”result is : ”+result);  
 uiHandler.sendMessage(msg);  
 }  
});

1. 通过内部类的方式（比如在一个Activity中）定义一个广播，该广播类需要定义成静态static类型的，不然会报 找不到空构造方法错误：

can't instantiate class cn.msyx.coolweather.activity.WeatherActivity$MyWeatherIdRestedReceiver; no empty constructor

*/\*\*  
 \* 接收当weatherId发生变化的广播消息的广播接收者  
 \*/* public static class MyWeatherIdRestedReceiver extends WeatherIdResetedReceiver {  
  
// @Override  
// public void refreshWeather(String weatherId) {  
//// Toast.makeText(MyApplication.getAppContext(), "MyWeatherIdRestedReceiver refreshWeather", Toast.LENGTH\_SHORT).show();  
// }  
  
 @Override  
 public boolean refreshWeatherData(String weatherId) {  
 Toast.*makeText*(MyApplication.*getAppContext*(), "Weather info has been refreshed!", Toast.*LENGTH\_SHORT*).show();  
 return true;  
 }  
 }