Demo20-动态注册广播的简单Demo

1. 创建一个类（或者Activity里的内部类），extends BroadcastReceiver, 这是广播接收器结果处理类，之后需要new一个对象注册广播接收器，给系统自己监听回调处理类的API：onReceive（）方法
2. 为了回收对象，不造成内存泄漏内存（内存回收不了），需要在Activity的onDestroy()方法中取消注册（猜测里面是直接回收这个处理类对象）

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.content.IntentFilter;

import android.net.ConnectivityManager;

import android.net.NetworkInfo;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private IntentFilter mIntentFilter;

private NetWorkChangeReceiver mNetWorkChangeReceiver;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//构造IntentFilter，指定广播接收器的响应值

mIntentFilter=new IntentFilter();

mIntentFilter.addAction("android.net.conn.CONNECTIVITY\_CHANGE");

//构造广播接收器处理类对象，给系统回调

mNetWorkChangeReceiver=new NetWorkChangeReceiver();

registerReceiver(mNetWorkChangeReceiver,mIntentFilter);

}

@Override

protected void onDestroy() {

super.onDestroy();

//回收内存，防止内存泄漏

unregisterReceiver(mNetWorkChangeReceiver);

}

//广播接收器结果处理类，对象注册后就给系统自己回调此类的API：onReceive

class NetWorkChangeReceiver extends BroadcastReceiver{

@Override

public void onReceive(Context context, Intent intent) {

//构造网络监听对象NetworkInfo

ConnectivityManager connectivityManager=(ConnectivityManager)getSystemService(Context.CONNECTIVITY\_SERVICE);

NetworkInfo networkInfo=connectivityManager.getActiveNetworkInfo();

if (networkInfo!=null&&networkInfo.isAvailable()){

Toast.makeText(context,"网络可用",Toast.LENGTH\_SHORT).show();

}else {

Toast.makeText(context,"网络不可用，请检查网络情况",Toast.LENGTH\_SHORT).show();

}

}

}

}