一、在清单文件中**注册静态广播接收（接收系统发出的广播）**

1、创建自定义类继承 extends BroadcastReceiver

2、在onReceive(Context context, Intent intent)方法中写收到广播要做的事，可以获取动作通过动作判断来处理相应的事

```
@Override
public void onReceive(Context context, Intent intent) {
        处理想做的事····
       //获取广播的类型或动作
       String action = intent.getAction();
       if("android.intent.action.MEDIA_MOUNTED".equals(action)){
            System.out.println("SD卡安装了");
           Toast.makeText(context,"SD卡安装了",0).show();
       }else if("android.intent.action.MEDIA_UNMOUNTED".equals(action)){
           System.out.println("SD卡卸载了");
           Toast.makeText(context,"SD卡卸载了",0).show();
         }
}
```

3、在清单文件中注册广播接收

```
<receiver android:name="cn.itheima.receiver.IpReceiver">   //自定义广播类 《必须》
  <intent-filter>      <intent-filter android:priority="1000">   接收广播的优先级   《可选》
            <!--
            action:name表示接收广播的动作    接收短信电话都是要相应权限的
            android.intent.action.NEW_OUTGOING_CALL:外拨电话的广播
            -->
           <action android:name="android.intent.action.NEW_OUTGOING_CALL"/>    //注册要接收什么广播的动作    《必须》

                <data android:scheme="file"/>  有时候会用到数据     《可选》

      </intent-filter>
</receiver>
```

二、自定义广播的接收与发送广播的方法：（**注册静态广播接收　接收自定义的广播**）

这种自定义广播接收者一个要自己写一个广播发送的方法来发送广播

1、创建自定义类继承 extends BroadcastReceiver

2、在onReceive(Context context, Intent intent)方法中写收到广播要做的事，可以获取动作通过动作判断来处理相应的事

3、在清单文件中注册广播接收者

```
<!-- 注册广播接收者 -->
    <receiver android:name="cn.itheima.receiver.MyReceiver">    自定义广播接收者类
        <intent-filter android:priority="1000">              接收广播的级别
            <action android:name="www.cctv.com"/>      接收广播的动作
        </intent-filter>
    </receiver>
```

4、创建广播发送者，发送自定义的广播

```
//创建意图
Intent intent = new Intent();
//设置广播的动作
intent.setAction("www.cctv.com");
//将意图中绑定数据
intent.putExtra("NAME","中央电视台");
//发送广播
this.sendBroadcast(intent);
```

三、无序广播和有序广播的发送，及发送（局部广播）只允许本应用内接收。

自定义广播接收者获取广播发送者的通过Intent 给的数据

@Override 第一个自定义广播接收者接收处理数据

```
public void onReceive(Context context, Intent intent) {

    //如果前面没有接收者的话，只有发送者的话，用如下代码获取意图中的内容
```

int money = intent.getIntExtra("MONEY",0); //获取通过intent 发送的数据

money-=200;

this.setResultData(money+""); //将改动的数据发送给后续接收者

//截断

this.abortBroadcast();

<第二个自定义广播接收者接收处理数据

在onReceive(Context context, Intent intent)方法中获取上一个接收者的数据

String strMoney = this.getResultData(); //后续接收者获取上一个接收者的数据

>

}

1、发送无序广播

```
Intent intent = new Intent();
intent.setAction("www.money.com");
intent.putExtra("MONEY",1000);
```

this.sendBroadcast(intent);

2、发送有序广播

```
Intent intent = new Intent();
intent.setAction("www.money.com");
intent.putExtra("MONEY",1000);//参数二：接收者的权限，如果写null的话，会优先级的高低依次接收
```

this.sendOrderedBroadcast(intent,null);

3、发送局部广播 只能够动态注册和取消

发送：

```
LocalBroadcastManager lbm = LocalBroadcastManager.getInstance(this);
Intent intent = new Intent();
intent.setAction("www.baidu.com");
mLbm.sendBroadcast(intent);
```

注册：

```
LocalBroadcastManager lbm = LocalBroadcastManager.getInstance(this);
ASASA mReceiver = new ASASA();  自定义广播接收者类
IntentFilter filter = new IntentFilter();
filter.addAction("www.baidu.com");
lbm.registerReceiver(mReceiver, filter);
```

反注册：

```
@Override
protected void onDestroy() {
    //取消广播接收者
    if(mReceiver!=null){
        mLbm.unregisterReceiver(mReceiver);
    }
    super.onDestroy();
}
```

接收：

```
public class ASASA extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context,"我真的收到了  不信你看",Toast.LENGTH_SHORT).show();
```

```
        }
    }
```

4、intent 指定包名Intent.setPackage设置广播仅对本程序有效

```
Intent intent = new Intent();
intent.setPackage("应用包名");
intent.setAction("消息action");
sendBroadcast(intent);
```

5、指定某一个receiver允许接收

```
Intent intent = new Intent();
intent. setComponent(newComponentName("包名", "Receiver类名"));
intent.setAction("消息action");
sendBroadcast(intent);
```

6、指定多个应用有权接收

AndroidManifest.xml

```
<permission android:description="@string/XXX"
      android:label="XXXX"
       android:name=" com.test.permission"
       android:protectionLevel=" signature">
<receiver
        android:name="XXXX"
        android:permission="com.test.permission" >
        <intent-filter >
            <actionandroid:name="XXXXX" />
        </intent-filter>
 </receiver>

Intent intent = new Intent();
intent.setAction("消息action");
sendBroadcast(intent, "com.test.permission");
```

四、有序广播的接收者可以对广播进行拦截篡改等操作

1、在接收到广播的自定义类里拦截广播 onReceive(Context context, Intent intent)方法中写

```
//截断广播   this.abortBroadcast();
```

自定义广播接收者获取广播发送者的通过Intent **给的数据**

```
@Override  第一个自定义广播接收者接收处理数据
public void onReceive(Context context, Intent intent) {
  //如果前面没有接收者的话，只有发送者的话，用如下代码获取意图中的内容
  int money = intent.getIntExtra("MONEY",0);      //获取通过intent 发送的数据
  money-=200;
  this.setResultData(money+""); //将改动的数据发送给后续接收者

  //截断
  this.abortBroadcast();

  <第二个自定义广播接收者接收处理数据
  在onReceive(Context context, Intent intent)方法中获取上一个接收者的数据
```

```
        String strMoney = this.getResultData();    //后续接收者获取上一个接收者的数据
    >
}
```

五、通过代码动态注册广播接收者

1、创建自定义类继承extends BroadcastReceiver

2、重写 onReceive(Context context, Intent intent)方法 处理收到广播要做的事，可以获取动作通过动作判断来处理相应的事

3、在需要接收广播的地方通过代码手动注册广播 并 注意反注册

```
ASASA  mReceiver = new ASASA();  自定义广播类
IntentFilter filter = new IntentFilter();
filter.addAction("www.baidu.com");     自定义广播的动作
this.registerReceiver(mReceiver,filter);     注册普通广播接收者
```

反注册：

```
@Override
protected void onDestroy() {
    //取消广播接收者
    if(mReceiver!=null){
        this.unregisterReceiver(mReceiver);
    }
    super.onDestroy();
}
```

4、由于上面是自定义广播接收者，所以需要手动发送一个广播

```
Intent intent = new Intent();
intent.setAction("www.baidu.com");
this.sendBroadcast(intent);
```

系统操作	action
监听网络变化	android.net.conn.CONNECTIVITY_CHANGE
关闭或打开飞行模式	Intent.ACTION_AIRPLANE_MODE_CHANGED
充电时或电量发生变化	Intent.ACTION_BATTERY_CHANGED
电池电量低	Intent.ACTION_BATTERY_LOW

电池电量充足（即从电量低变化到饱满时会发出广播	Intent.ACTION_BATTERY_OKAY
系统启动完成后(仅广播一次)	Intent.ACTION_BOOT_COMPLETED
按下照相时的拍照按键(硬件按键)时	Intent.ACTION_CAMERA_BUTTON
屏幕锁屏	Intent.ACTION_CLOSE_SYSTEM_DIALOGS
设备当前设置被改变时(界面语言、设备方向等)	Intent.ACTION_CONFIGURATION_CHANGED
插入耳机时	Intent.ACTION_HEADSET_PLUG
未正确移除SD卡但已取出来时(正确移除方法:设置 - SD卡和设备内存 - 卸载SD卡)	Intent.ACTION_MEDIA_BAD_REMOVAL
插入外部储存装置（如SD卡）	Intent.ACTION_MEDIA_CHECKING
成功安装APK	Intent.ACTION_PACKAGE_ADDED
成功删除APK	Intent.ACTION_PACKAGE_REMOVED
重启设备	Intent.ACTION_REBOOT
屏幕被关闭	Intent.ACTION_SCREEN_OFF
屏幕被打开	Intent.ACTION_SCREEN_ON
关闭系统时	Intent.ACTION_SHUTDOWN
重启设备	Intent.ACTION_REBOOT

* GPRS 2G(2.5) General Packet Radia Service 114kbps
* EDGE 2G(2.75G) Enhanced Data Rate for GSM Evolution 384kbps
* UMTS 3G WCDMA 联通3G Universal MOBILE Telecommunication System 完整的3G移动通信技术标准
* CDMA 2G 电信 Code Division Multiple Access 码分多址
* EVDO_0 3G (EVDO 全程 CDMA2000 1xEV-DO) Evolution - Data Only (Data Optimized) 153.6kps - 2.4mbps 属于3G
* EVDO_A 3G 1.8mbps - 3.1mbps 属于3G过渡，3.5G
* 1xRTT 2G CDMA2000 1xRTT (RTT - 无线电传输技术) 144kbps 2G的过渡,
* HSDPA 3.5G 高速下行分组接入 3.5G WCDMA High Speed Downlink Packet Access 14.4mbps
* HSUPA 3.5G High Speed Uplink Packet Access 高速上行链路分组接入 1.4 - 5.8 mbps
* HSPA 3G (分HSDPA,HSUPA) High Speed Packet Access
* IDEN 2G Integrated Dispatch Enhanced Networks 集成数字增强型网络 （属于2G，来自维基百科）
* EVDO_B 3G EV-DO Rev.B 14.7Mbps 下行 3.5G
* LTE 4G Long Term Evolution FDD-LTE 和 TDD-LTE , 3G过渡，升级版 LTE Advanced 才是4G
* EHRPD 3G CDMA2000向LTE 4G的中间产物 Evolved High Rate Packet Data HRPD的升级
* HSPAP 3G HSPAP 比 HSDPA 快些

```
//获取网络连接管理者
ConnectivityManager connectionManager = (ConnectivityManager)context.getSystemService(Context.CONNECTIVITY_SERVI
//获取网络的状态信息，有下面三种方式
NetworkInfo networkInfo = connectionManager.getActiveNetworkInfo();
```

getDetailedState()：获取详细状态。

getExtraInfo()：获取附加信息。

getReason()：获取连接失败的原因。

getType()：获取网络类型(一般为移动或Wi-Fi)。

getTypeName()：获取网络类型名称(一般取值“WIFI”或“MOBILE”)。

isAvailable()：判断该网络是否可用。

isConnected()：判断是否已经连接。

isConnectedOrConnecting()：判断是否已经连接或正在连接。

isFailover()：判断是否连接失败。

isRoaming()：判断是否漫游

当用wifi上的时候

getType 是WIFI

getExtraInfo是空的

当用手机上的时候

getType 是MOBILE

用移动CMNET方式

getExtraInfo 的值是cmnet

用移动CMWAP方式

getExtraInfo 的值是cmwap 但是不在代理的情况下访问普通的网站访问不了

用联通3gwap方式

getExtraInfo 的值是3gwap

用联通3gnet方式

getExtraInfo 的值是3gnet

用联通uniwap方式

getExtraInfo 的值是uniwap

用联通uninet方式

getExtraInfo 的值是uninet

网络改变的广播监听.java