



## 实验五

### appwidget 及 broadcast 使用

---

## 【实验目的】

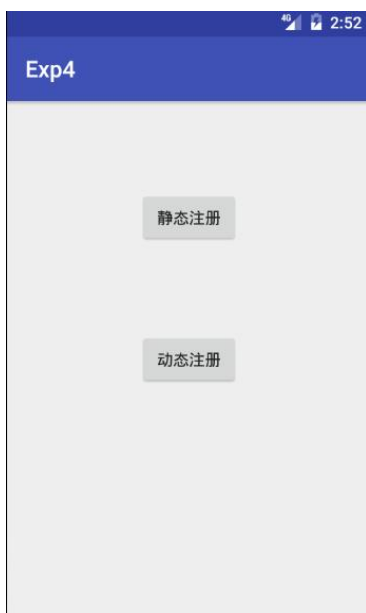
- 1、掌握 AppWidget 编程基础
- 2、掌握 Broadcast 编程基础
- 3、掌握动态注册 Broadcast 和静态注册 Broadcast

## 【实验内容】

实现一个 Android 应用，实现静态广播、动态广播两种改变 widget 内容的方法。在上次实验的基础上进行修改，所以一些关于静态动态广播的内容会简略。

具体要求：

- (1) 该界面为应用启动后看到的界面。



widget 初始情况如下：



(2) 点击静态注册按钮，跳转至如下界面。



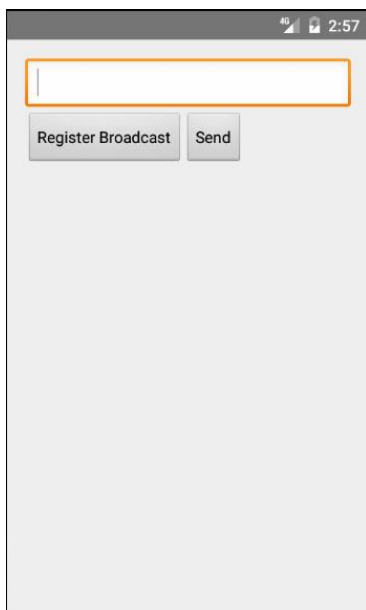
点击表单项目。如 banana.widget 会发生对应变化。点击 Widget 上的图片可以跳转回主页面



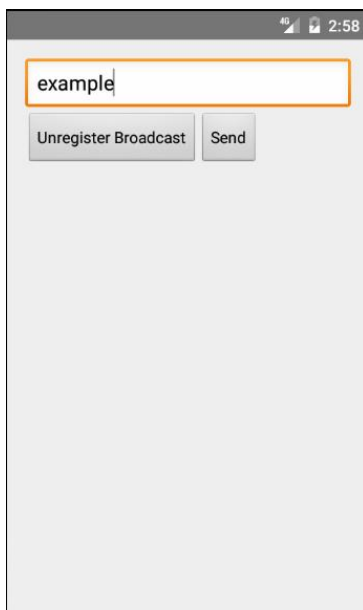
(3) 点击动态注册按钮，跳转至如下界面。

实现以下功能：

- a) 可以编辑广播的信息，点击 Send 按钮发送广播。
- b) 设置一个按钮进行广播接收器的注册与注销。
- c) 广播接收器若已被注册，发送出的广播信息能够及时更新桌面上 Widget 上文字内容及更新为默认 dynamic 图片。
- d) 点击 Widget 上的图片可以跳转回主页面。



（未注册广播）



（已注册广播）

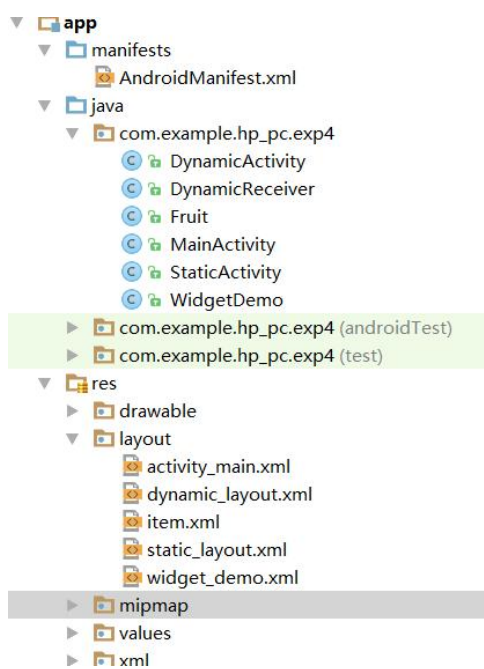


（已注册后点击发送）

注：在设置按钮内容的时候注意大小写问题。

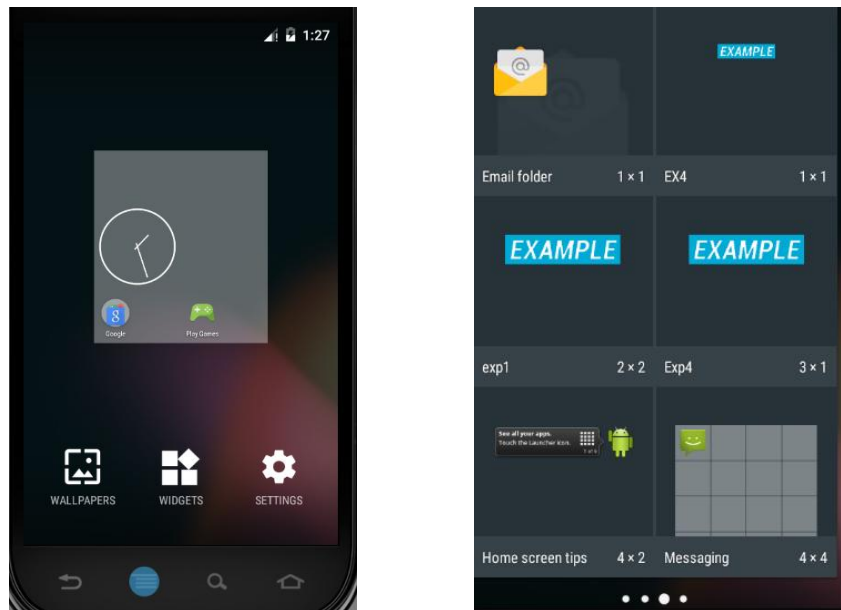
## 【参考内容】

（1）参考文件目录如下：

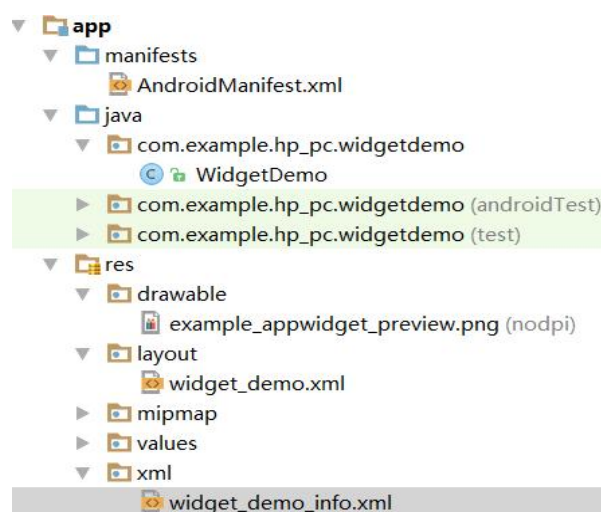


## (2) Widget

widget 的添加：长按菜单键，点击 widgets 选项。找到对应的 widget 将其拖入桌面。对于不同的 API 版本显示会稍有不同。



典型的 Android Widget 有三个主要组件，一个边框、一个框架和图形控件以及其他元素。在 Android Studio 中创建 Widget 类后，会直接生成相关文件。



---

1. Widget 布局文件 widget\_demo.xml，布局中有一个 ImageView，一个 TextView。

要求：文字颜色为红色，大小为 20dp，整体背景为透明。最后效果如下：



2. Widget 内容提供者文件 widget\_demo\_info.xml，编辑该文件，设置其大小属性和布局，如下图：

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialKeyguardLayout="@layout/widget_demo"
    android:initialLayout="@layout/widget_demo"
    android:minHeight="55dp"
    android:minWidth="200dp"
    android:previewImage="@drawable/example_appwidget_preview"
    android:resizeMode="horizontal|vertical"
    android:updatePeriodMillis="86400000"
    android:widgetCategory="home_screen"></appwidget-provider>
```

其中，minWidth 为最小宽度，minHeight 为最小高度，initialLayout 为初始布局。

3. 修改 WidgetDemo.java 代码，重写 onUpdate 方法，为 Widget 添加事件，使得能够返回主页面。

这里需要使用到一种用户程序访问主屏幕和修改特定区域内容的方法：**RemoteView 架构**。RemoteView 架构允许用户程序更新主屏幕的 View，点击 Widget 激活点击事件，Android 会将其转发给用户程序，由 AppWidgetProviders 类处理，使得用户程序可更新主屏幕 Widget。

```
super.onUpdate(context, appWidgetManager, appWidgetIds);
Intent clickInt = new Intent(context, MainActivity.class);
PendingIntent pi = PendingIntent.getActivity(context, 0, clickInt, 0);
RemoteViews rv = new RemoteViews(context.getPackageName(),
    R.layout.widget_demo);
rv.setOnClickPendingIntent(R.id.widget_image, pi);
appWidgetManager.updateAppWidget(appWidgetIds, rv);
```

**pendingIntent** 是一种特殊的 Intent。主要的区别在于 Intent 的执行立刻的，而 pendingIntent 的执行不是立刻的。本次使用方法类的静态方法为 getActivity(Context, int, Intent, int)，对应 Intent 的跳转到一个 activity 组件的操作。

---

### (3) 静态广播部分

静态广播部分首先需要实现一个 listView 的显示，这一部分在实验三中有详细说明这里不加赘述。之后设置 setOnItemClickListener，利用 bundle 和 intent 将图片与文字内容发送出去。

```
Intent intent = new Intent(STATICACTION); // 定义Intent
intent.putExtras(bundle);
sendBroadcast(intent);
```

参考代码中的 STATICACTION 为自己设定的广播名称。由于是静态注册所以需要在 AndroidManifest.xml 中进行注册。

```
<receiver android:name=".WidgetDemo">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="com.example.exp4.staticreceiver" />
    </intent-filter>

    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_demo_info" />
</receiver>
```

在 Widget 类中重写 onReceive 方法，这里需要使用到 RemoteView 以及 Bundle。当接收到对应广播时进行数据处理。

```
@Override
public void onReceive(Context context, Intent intent) {
    super.onReceive(context, intent);
    RemoteViews rv = new RemoteViews(context.getPackageName(),
        R.layout.widget_demo);
    Bundle bundle = intent.getExtras();

    if (intent.getAction().equals(STATICACTION)) {

    }
}
```

if 条件语句中主要用到的函数为：setTextViewText、setImageViewResource。之后使用 AppWidgetManager 类对 Widget 进行更新。



#### (4) 动态广播部分

1. 实现 BroadcastReceiver 子类（这里命名为 DynamicReceiver），并且重写 onReceive 方法，修改方法与 widget 类中类似。

```
public class DynamicReceiver extends BroadcastReceiver {  
    private static final String DYNAMICACTION = "com.example.exp4.dynamicreceiver"; //动态广播的Action字符串  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        if (intent.getAction().equals(DYNAMICACTION)) { //动作检测  
            RemoteViews rv = new RemoteViews(context.getPackageName(),  
                R.layout.widget_demo);  
            Bundle bundle = intent.getExtras();  
            //TODO:修改widget的文字与图片  
        }  
    }  
}
```

2. 创建主页面，设置注册和注销按钮(set OnClickListener)，使其能够注册或者注销广播器。

注册关键代码：

```
IntentFilter dynamic_filter = new IntentFilter();  
dynamic_filter.addAction(DYNAMICACTION); //添加动态广播的Action  
registerReceiver(dynamicReceiver, dynamic_filter); // 注册自定义动态广播消息
```

注销关键代码：

```
unregisterReceiver(dynamicReceiver);
```

其中 dynamicReceiver 为我们之前创建的 DynamicReceiver 类。用 registerReceiver 与 unregisterReceiver 分别对其进行注册与注销。



---

## 【检查内容】

- 1、布局显示是否正常。
- 2、静态广播：点击表单项 widget 是否会发生对应变化。
- 3、动态广播：点击注册后发送内容 widget 是否会发生对应变化。注销或未注册时，要求 widget 不会发生变化。
- 4、点击 widget 的 Image 是否会正确返回主界面。

注：demo 中有 notification 的部分在实际运行时可以不添加。

## 【提交说明】

- 1、deadline：下一次实验课前一天晚上 12 点
- 2、提交位置：<ftp://222.200.185.18:1890/> 对应文件夹下
- 3、命名与目录要求：

附件命名及格式要求：学号\_姓名\_labX.zip（姓名中文拼音均可）

重复提交命名格式要求：学号\_姓名\_labX\_Vn.zip

目录结构：

```
14331111_huashen_lab1 --  
|  
-- lab1实验报告.pdf  
|  
-- lab1_code (包含项目代码文件)
```

其中项目代码文件为项目文件夹，提交之前先 clean