

# Android开发者不得不知的13个注意事项

作者: [balala](#)

原文链接: <https://ld246.com/article/1450866392725>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在安卓开发中，相信大家都会遇到各种各样的问题，掌握一些开发技巧，并有效的避免这些坑还是灰重要的。Android老鸟们应该在夜以继日的码农生涯中总结了不少坑和避开这些坑的方法，菜鸟们就比较可怜了，刚踏入码农界，一无所知，常常被这些坑伤滴遍体鳞伤。本文是小编在android开发教程上搜罗的一些android开发中比较常见的坑和解决办法，菜鸟们都看过来吧

---

### 1. 在Activity还没完全显示时，弹出PopupWindow或者Dialog 崩Activity not running 错误

解决方案：重写onWindowFocusChanged方法

```
public void onWindowFocusChanged(boolean hasFocus) {  
  
    if (hasFocus) {  
        if (!mIsInitData) {  
            initData();  
            mIsInitData = true;  
        }  
    }  
  
    super.onWindowFocusChanged(hasFocus);  
  
}  
  
protected void initData() {  
    // 在此处编写弹出Popup或者Dialog的方法  
}
```

### 2. 在library中使用switch语句处理id 时报错

产生原因：library中生成的R文件中生成的id 没有用final 修饰；

解决方案：使用if else 替换switch

### 3. 不要在Application中缓存任何数据， NoPointException

产生原因：一般情况下不会出现异常，当按下Home 应用隐藏到后台，长时间未使用，导致应用被回，当再次启动时， Application会重新创建，而Activity此时再向Application里取数据，异常发生

解决方案：将数据缓存到sp 或者数据库或者sd卡。或者在使用Application缓存数据时，进行null判断

### 4. AsyncTask只能执行一次

### 5. 提取一个BaseActivity， 里面进行一些统一处理，能让你的代码更简洁。

```
public abstract class BaseActivity extends Activity implements OnClickListener {  
  
    private boolean mIsInitData;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
}
```

```

@Override
public void setContentView(int layoutResID) {
    super.setContentView(layoutResID);
    initView();
}
@Override
public void onWindowFocusChanged(boolean hasFocus) {

    if (hasFocus) {
        if (!mIsInitData) {
            initData();
            mIsInitData = true;
        }
    }

    super.onWindowFocusChanged(hasFocus);
}

protected void initData() {

}

protected abstract void initView();

/**
 * 获取View
 *
 * @param id
 * @return
 */
@SuppressWarnings("unchecked")
protected <T extends View> T mGetView(int id) {
    return (T) findViewById(id);
}

/**
 * 获取Button的实例 并绑定点击事件
 *
 * @param id
 * @return
 */
protected Button mGetButtonSetOnClick(int id) {

    Button btn = (Button) findViewById(id);
    btn.setOnClickListener(this);
    return btn;
}

/**
 * 获取ImageView的实例 并绑定点击事件
 *
 * @param id
 * @return
 */

```

```

protected ImageView mGetImageViewSetOnClick(int id) {

    ImageView image = (ImageView) findViewById(id);
    image.setOnClickListener(this);
    return image;
}

/**
 * 获取View的实例 并绑定点击事件
 *
 * @param id
 * @return
 */
protected View mGetViewSetOnClick(int id) {

    View view = (View) findViewById(id);
    view.setOnClickListener(this);
    return view;
}

/**
 * 获取TextView中的文本信息
 *
 * @param tv
 * @return
 */
protected String mGetTextViewContent(TextView tv) {
    return tv.getText().toString().trim();
}

/**
 * 获取EditText中的文本信息
 *
 * @param et
 * @return
 */
protected String mGetEditTextContent(EditText et) {
    return et.getText().toString().trim();
}

protected void showHintMsg(int sid) {
    MToast.showToast(this, getResources().getString(sid));
}

protected void showHintMsg(String sMsg) {
    MToast.showToast(this, sMsg);
}

}

```

6. Toast定义为全局，避免一直不断的卡卡卡卡。

```

public class MToast {
private static Toast mToast;

```

```

private static TextView tv_content;

public static void showToast(Context context, String msg) {
    try {

        if (mToast == null) {
            mToast = Toast.makeText(context, msg, Toast.LENGTH_SHORT);
            mToast.setGravity(Gravity.TOP, 0,
                DensityUtil.dip2px(context, 3));
            View view = View.inflate(context, R.layout.m_toast, null);
            tv_content = (TextView) view.findViewById(R.id.tv_content);
            mToast.setView(view);
            tv_content.setText(msg);
        } else {
            tv_content.setText(msg);
        }
        mToast.show();
    } catch (Exception e) {
        // TODO: handle exception
    }
}
}
}

```

7. 标题栏样式抽取，抽取思路大概有两种：

第一种：用 `<include>` 标签在xml布局时引入

第二种：自定义一个TitleView。

不仅仅是标题栏，字体大小，主题颜色，能抽取的都统一处理。

8. `TextView.setText()`；中要显示int类型的值，用`String.valueOf()`转，不要直接`124+ ""`，不知道什么这样的童鞋，去看看源码就知道为什么了。

9. 退出应用方式：

1) 直接杀死进程

2) 在BaseActivity中注册一个广播，发送广播关闭

3) 定义一个全局容器存储Activity应用，退出时遍历退出（不推荐）

10. 一个功能分几个页面处理时，使用Dialog 模拟Activity 避免了数据在Activity之间传递。

11. 手机重启，通过不断的new 空Toast，导致系统奔溃而重启，想想竟有一种无言以对的感觉，原来oast还可以这么玩

```

public void onClick(View v){
    while(true){
        Toast toast = new Toast(this);
        toast.setView(new View(this));
        toast.show();
    }
}
}

```

12. View类中的`setSelected(boolean)`方法结合`android:state_selected=""` 用来实现图片选中效果

自定义标题栏用起来很方便;

13. EditText 中有个 android:digits="" 属性, 用来自定义输入的字符类型, 比如输入身份证只能是字和x或者X 使用 android:digits="1234567890xX" 就轻松搞定了, 不用再在代码里面进行校验了;

以上就是android中, 比较常见的一些问题, 及出现这些问题的原因、解决办法, 可能还有整理得不很全面的地方, 欢迎大家补充分享。

相关文章: 《Android学习之JSON解析详解》

<http://www.maiziedu.com/group/article/10955/>