Android开发参考手册

版本号：0.1

创建日期：2017-03-02

最后更新：2017-03-02

作者：李晓龙 [lixiaolong\_cpu@sari.ac.cn](mailto:zhangy_cpu@sari.ac.cn)



版本历史[1]

| 日期 | 版本号 | 作者 | 修改记录 |
| --- | --- | --- | --- |
| 2016-08-01 | 0.1 | 李晓龙 | 初始文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1]: 版本历史由文档修订人负责填写。

审批历史[2]

| 审批人 | 邮件 | 角色 | 版本号 | 审批意见 |
| --- | --- | --- | --- | --- |
| 李晓龙 | [lixiaolongshanghai@sari.ac.cn](mailto:lixiaolongshanghai@sari.ac.cn) | BMC负责人 | 0.1 | 同意 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

[2]: 审批历史由文档审阅人负责填写。

免责声明

版权所有 ©2016中科院通用芯片与基础软件研究中心。本中心保留所有权利。未经中科院通用芯片与基础软件研究中心事先书面许可， 严禁本出版物的任何部分被复制，传递，转录，存储于检索系统，或翻译成任何语言或计算机语言。 严禁以任何形式或通过任何手段，包括但不限于电子，机械，磁性，光学，化学，手动达到以上目的。

目录

免责声明 3

目录 4

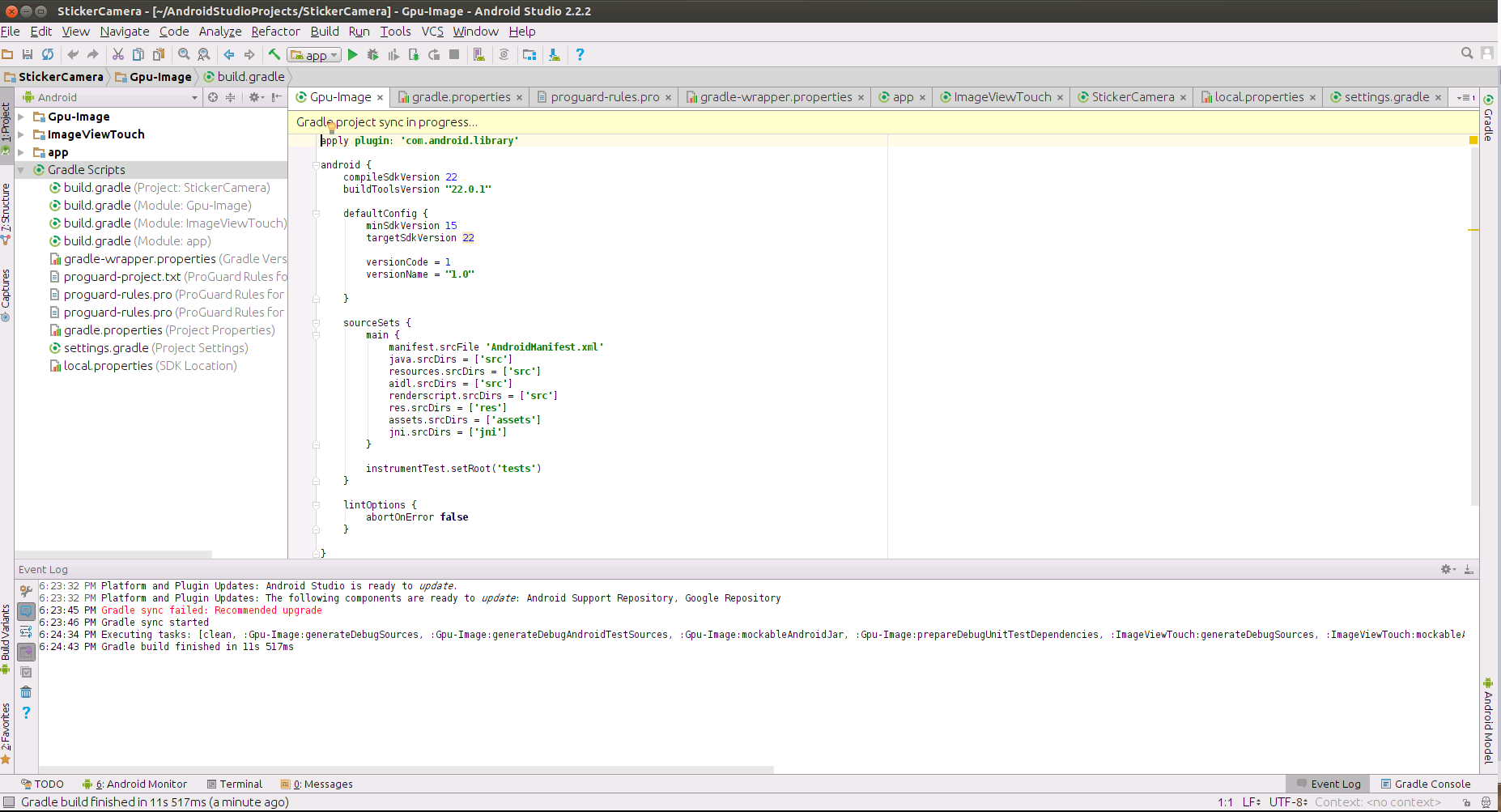
1 HTML标签 5

2 HTML5标签 7

3 CSS属性 9

4 CSS3属性 11

# Android Studio



Android Studio为谷歌开发的为了帮助开发者开发谷歌程序的IDE工具集，类似BMC的MDS工具，也是基于Eclipse架构构建的。

常用的使用技巧包括：

ctrl + 空格：代码提示

ctrl + q : 显示类的文档。

Alt + insert: 快速插入getter and setter等类函数。

# Android应用架构

Android的应用架构包括如下的文件夹：

* app
* build
* gradle

## build.gradle : 用于控制build流程

buildscript {

} //设置脚本的运行环境。

1. repositories：支持java依赖库管理，用于项目的依赖。

2. dependencies：依赖包的定义。

3. apply plugin：声明构建的项目类型，通常是android。

4. android：设置编译android项目的参数。

5. signingConfigs：签名相关的配置。

## 其他文件

settings.gradle - 告诉Android Studio相关的可以进一步使用的文件夹

其中的关键字include用于展示所有包含的应用模组。

gradle.properties - 设置一些属性常量。

gradlew - 就是gradle wrapper，

gradlew.bat - 要执行的文件程序。使用该脚本可以在没有安装gradle的系统中做build。

gradle-wrapper - wrapper task，用于在没有安装gradle的系统中做build。

local.properties - 本地的属性。

\*.iml - intellj的项目文件。

## androidManifest.xml文件

在app目录下存在一个androidManifest.xml文件。

在Android下面存在四大组件：

* [Activity](http://www.ourunix.org/android/post/59.html" \t "http://blog.csdn.net/weiguang_123/article/details/_blank)
* [BroadCast receiver](http://www.ourunix.org/android/post/70.html" \t "http://blog.csdn.net/weiguang_123/article/details/_blank)
* [service](http://www.ourunix.org/android/post/73.html" \t "http://blog.csdn.net/weiguang_123/article/details/_blank)
* [Content Provider](http://www.ourunix.org/android/post/78.html" \t "http://blog.csdn.net/weiguang_123/article/details/_blank)

程序中如果用到这四大组件需要在androidManifest中声明。

## android调试方法

Android的常用调试方法有：

1. Toast
2. Alert

android的默认继承级别：



# Android系统开发基础

应用的框架如下：

Views - 可扩展的视图Views。

内容管理器（Content Providers）- 使得应用程序可以访问另一个应用程序的数据，或者共享它们自己的数据。

资源管理器（Resource Manager）- 提供非代码资源的访问。

通知管理器（Notification Manager）- 使得应用程序可以在状态栏中显示客户通知信息。

活动管理器（Activity Manager）- 用来管理应用程序生命周期并提供导航和回退功能。

Android系统由Activity、Service、Broadcast Receiver和Content Provider组成。其中Activity是使用频率最高、最重要的组件。

## Android资源

在Android中，使用@号来引用资源，在@号后面依次输入类型（id）和资源名称。

只有第一次定义某个资源的id时才需要使用一个加号’+’。

在XML中，使用android:id属性来定义一个资源。

android:layout\_width

/android:layout\_height

而res/values/strings.xml中定义了所有使用到的字符串信息。

使用weight属性来完成空间的物理布局：

所有视图的默认 weight 值都为 0，所以如果您仅将一个视图的 weight 值指定为大于 0，那么等到其他所有视图都获得所需空间后，该视图便会填满所有剩余空间。

## Activity活动单元

多个Activity组成了活动栈（Stack），当前活动的Activity位于栈顶。复写Activity的onCreate来完成创建的内容。

从一个Activity跳转到另外一个Activity的方法：startActivity() 或者 startActivityForResult()方法。

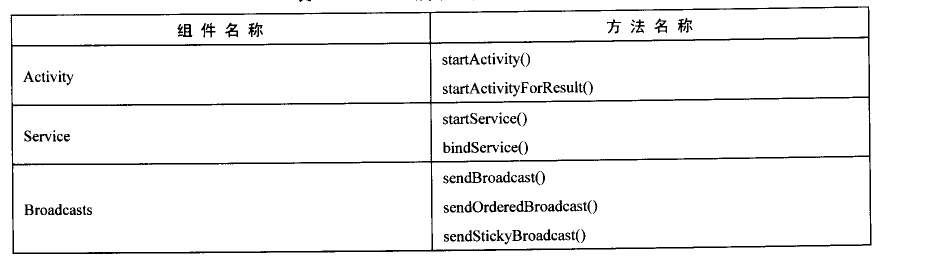
Activity生命周期：

1. onCreate()
2. 可见生命周期onStart() -> onStop()
3. 前台生命周期onResume() -> onPause()结束。

## Activity的消息传递

把要传递的消息封装在intent对象里面，通过intent对象传递到另一个Intent中。

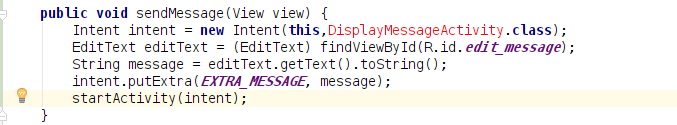
Intent就像一个媒婆。Intent可以启动一个Activity，也可以启动一个Service，还可以发起一个广播（Broadcast）



Intent主要由以下几个部分组成：

动作（Action）、数据（Data）、分类（Category）、类型（Type）、组件（Component）和扩展信息（Extra）。

Intent在寻找目标时有两种方法：（1）通过组件名称直接指定；（2）通过Intent Filter过滤指定。



有多少个Activity将会被定义在AndroidManifest.xml文件中！

## Service组件

Service就是运行在后台的一种服务程序，一般很少和用户交互，因此没有可视化界面。

实现Service类只要继承Service类，然后实现其生命周期中的方法即可。

startService() : 启动一个Service或者使用bindService方法来绑定一个存在的Service。

onBind: 返回一个绑定的接口给Service。

onBind()：必须实现的一个方法，返回一个绑定的接口给Service。

onCreate(): 当Service第一次被创建时，由系统调用。

onStart(): 当通过startService()方法启动Service时，该方法被调用。

onDestroy()：当Service不再使用时调用。

## Broadcast Receiver

一个应用可以通过sendBroadcast（）方法来发起一个系统级别的事件广播来传递消息。可以在应用程序中实现Broadcast Receiver来监听和响应这些广播的Intent。

# Android开发实例

## Fragment活动UI

# 开源项目1 - Stickercamera

## 代码解析

Android中的资源访问 - res目录，该目录中包含了所有可见的资源文件。

包括：

xml动画资源

xml字符串资源（string）

xml菜单资源（menu）

xml布局文件（layout）

xml数组文件

xml颜色文件（color）

xml尺寸文件（dimen）

xml样式文件

drawable资源文件

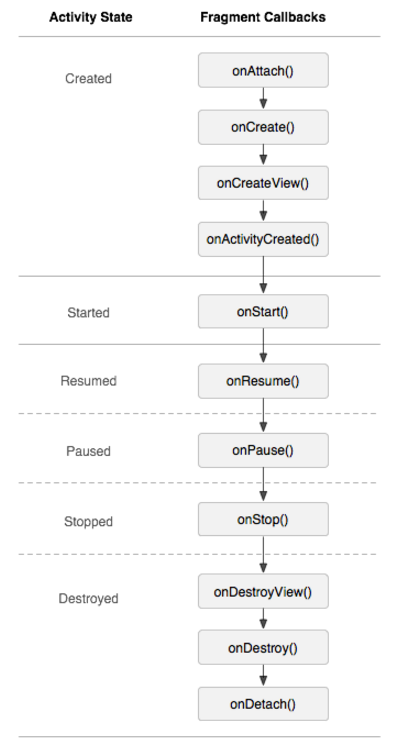
## RelativeLayout

RelativeLayout的特点是元素的位置不用固定，可以指定相对位置。

## MainActivity

在OnCreate函数中，使用setContentView()接口设置R.layout.activity\_main，自动绑定到activity\_main.xml布局文件中。

Activity和Fragment的生命周期图如下所示：



其中，Fragment比Activity增加了几个额外的生命周期接口：

1. onAttach() : 当fragment与Activity发生关联时调用。
2. onCreateView()：创建该Fragment的视图。
3. onActivityCreated(Bundle)
4. onDestroyView()，当Fragment视图被移除时调用。
5. onDetach()：与onAttach对应。

Fragment就是为了包含特定的View而采用的。

# 附录-常用类成员

|  |  |  |
| --- | --- | --- |
| 类名 | 主要功能 | 说明 |
| ButterKnife | 解放findViewById() | 在setContentView()之后调用 |
| Fragment | 创建子View | 用于适应不同型号的窗体大小 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 附录 - 常用快捷键

* Alt + Enter快捷键完成了自动import类型的功能。
* 快捷键“ctrl+q”显示说明文档。