Android客户端架构及公共机制

说明文档

中兴通讯股份有限公司

IT中心

2014年5月

**修订历史记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **日期** | **描述** | **版本** | **作者** | **审核者** |
| 1 | 2014-5-19 | 文档初建 | 0.1 | 王恩子 |  |
| 2 | 2014-5-25 | 修改组件说明 | 0.2 | 王恩子 |  |
| 3 | 2014-6-18 | 新增Validation | 0.3 | 王恩子 |  |
|  |  |  |  |  |  |

本文档中所包含的信息属于机密信息，如无中兴通讯股份有限公司的书面许可，任何人都无权复制或利用。

Copyright®2014 by ZTE Co. Ltd

**目录**

[1 引言 5](#_Toc403586195)

[1.1 编写目的 5](#_Toc403586196)

[1.2 文档对象 5](#_Toc403586197)

[1.3 术语约定 5](#_Toc403586198)

[2 架构图 5](#_Toc403586199)

[2.1 架构图描述 6](#_Toc403586200)

[3 业务应用开发目录参考 7](#_Toc403586201)

[3.1 业务功能包子目录参考 7](#_Toc403586202)

[4 开发依赖库工程 8](#_Toc403586203)

[5 示例工程 8](#_Toc403586204)

[6 组件说明 8](#_Toc403586205)

[6.1 ZTEAndroid-APP 8](#_Toc403586206)

[6.1.1 基本说明 8](#_Toc403586207)

[6.1.2 API介绍 9](#_Toc403586208)

[6.2 ZTEAndroid-Async-Http 14](#_Toc403586209)

[6.2.1 基本说明 14](#_Toc403586210)

[6.2.2 API介绍 15](#_Toc403586211)

[6.3 ZTEAndroid-Resource 15](#_Toc403586212)

[6.3.1 基本说明 15](#_Toc403586213)

[6.3.2 API介绍 15](#_Toc403586214)

[6.4 ZTEAndroid-ORM 15](#_Toc403586215)

[6.4.1 基本说明 15](#_Toc403586216)

[6.4.2 API介绍 16](#_Toc403586217)

[6.5 ZTEAndroid-EventBus 16](#_Toc403586218)

[6.5.1 基本说明 16](#_Toc403586219)

[6.5.2 原理介绍 16](#_Toc403586220)

[6.5.3 API介绍 18](#_Toc403586221)

[6.6 ZTEAndroid-LogMgr 18](#_Toc403586222)

[6.6.1 基本说明 18](#_Toc403586223)

[6.6.2 API介绍 18](#_Toc403586224)

[6.7 ZTEAndroid-Cache 19](#_Toc403586225)

[6.7.1 基本说明 19](#_Toc403586226)

[6.7.2 API介绍 19](#_Toc403586227)

[6.8 ZTEAndroid-Widget 19](#_Toc403586228)

[6.8.1 基本说明 19](#_Toc403586229)

[6.8.2 API介绍 20](#_Toc403586230)

[6.9 ZTEAndroid-AppUpdate 20](#_Toc403586231)

[6.9.1 基本说明 20](#_Toc403586232)

[6.9.2 API介绍 20](#_Toc403586233)

[6.10 ZTEAndroid-SecurityAuth 20](#_Toc403586234)

[6.10.1 基本说明 20](#_Toc403586235)

[6.10.2 API介绍 21](#_Toc403586236)

[6.11 ZTEAndroid-Validation 21](#_Toc403586237)

[6.11.1 基本说明 21](#_Toc403586238)

[6.11.2 API介绍 22](#_Toc403586239)

[6.12 ZTEAndroid-Common 23](#_Toc403586240)

[6.12.1 基本说明 23](#_Toc403586241)

[6.12.2 API介绍 23](#_Toc403586242)

[7 其他机制 24](#_Toc403586243)

[7.1 依赖注入 24](#_Toc403586244)

[7.1.1 原理介绍 24](#_Toc403586245)

[7.2 换肤机制 24](#_Toc403586246)

[7.2.1 原理介绍 24](#_Toc403586247)

[8 编码规约 25](#_Toc403586248)

[8.1.1 Java通用编码规约 25](#_Toc403586249)

[8.1.2 资源目录和资源文件规约 26](#_Toc403586250)

[8.1.3 中文处理规约 28](#_Toc403586251)

# 引言

## 编写目的

* 帮助相关负责人了解Android客户端开发框架整体方案。
* 为开发人员进行后续的开发提供指导。

## 文档对象

* 相关技术负责人及开发工程师。

## 术语约定

* 如无特殊说明，后面文档出现的“中兴”都是指“中兴通讯股份有限公司”。

# 架构图



图2-1 架构图1



图2-1 架构图2

## 架构图描述

* ZTEAndroid-APP：扩展Android自带组件，例如Activity、Fragment、Adapter等
* ZTEAndroid-Common：公共组件。
* ZTEAndroid-Resource：Android资源访问组件。
* ZTEAndroid-ORM：数据库ORM组件。
* ZTEAndroid-Async-Http：Http异步通讯组件。
* ZTEAndroid-EventBus：事件总线。
* ZTEAndroid-LogMgr：日志管理组件。
* ZTEAndroid-Cache：缓存组件，图片缓存，文件缓存。
* ZTEAndroid-Widget：可复用的自定义控件
* ZTEAndroid-Validation：UI校验框架，提高UI表单校验的效率
* ZTEAndroid-SecurityAuth：安全认证组件。
* ZTEAndroid-AppUpdate：应用升级组件。
* ZTEAndroid-Pay：客户端支付组件，快速接入支付宝及银联支付。

# 业务应用开发目录参考

## 业务功能包子目录参考

**com.zte.android.biz**

├─**biz1 目录**

│├─**activity -- 主控制器**

│├─**fragment -- UI碎片(子控制器)**

│├─**dialog -- 对话框**

│├─**appservice -- 业务层**

│├─**adapter -- UI和数据之间适配器**

│├─**dao -- 数据访问层**

│├─**model -- 数据模型**

│├─**event -- 自定义事件,由事件总线驱动**

│├─**helper -- 帮助类**

└─**enums -- 枚举**

├─**biz2** **目录**

│├─**activity**

│├─**fragment**

│├─**dialog**

│├─**appservice**

│├─**adapter**

│├─**dao**

│├─**model**

│├─**event**

│├─**helper**

└─**enums**

# 开发依赖库工程

具体的业务开发过程中，需要导入以下三个工程:

1. ZTEAndroid-App-Lib

集成了ZTEAndroid-Framework的各种子组件,该工程依赖ZTEAndroid-ResourceLib。业务应用只需要依赖该工程即可。

[http://10.88.144.141:9999/userContent/EMM-Android/发布版本/lib/1.1.5/ZTEAndroid-App-Lib-1.1.5.zip](http://10.88.144.141:9999/userContent/EMM-Android/发布版本/lib/1.1.0/ZTEAndroid-App-Lib-1.1.0.zip)

1. ZTEAndroid-ResourceLib

资源相关的库，业务应用可修改资源的值或者在自身的应用中同名覆盖资源值，但不要修改资源的名称或者ID。该工程依赖android-support-v7-appcompat。

[http://10.88.144.141:9999/userContent/EMM-Android/发布版本/lib/1.1.5/ZTEAndroid-ResourceLib-1.1.5.zip](http://10.88.144.141:9999/userContent/EMM-Android/发布版本/lib/1.1.0/ZTEAndroid-ResourceLib-1.1.0.zip)

1. Android官方的兼容库android-support-v7-appcompat。



# 示例工程



# 组件说明

## ZTEAndroid-APP

### 基本说明

扩展Android自带组件，例如Activity、Fragment、Adapter等，开发具体的业务应用时，主要依赖该组件库。

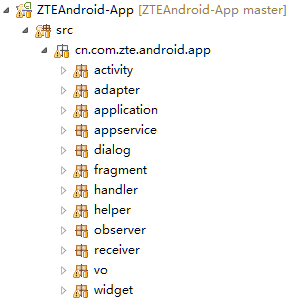


图3-1 App结构图

### API介绍

详情请参考API文档

#### Application基类

目的和用途：为所有APP提供Application基类支持。

技术标准：

* 1. 所有的业务APP的APP的Application都必须从框架提供的Application基类继承。
  2. 基类自动初始化：线程池、数据库管理器、资源管理器、Http通讯管理器、缓存管理器、文件存贮管理器、单点登陆认证文件管理器、Activity管理器、日志管理器



子类常用覆盖的方法：

| 方法名 | 方法作用 | 是否必须 | 补充说明 |
| --- | --- | --- | --- |
| initDBManager | 初始化数据库管理器 | 否 | 如果APP需要数据库存贮，则需要覆盖方法 |
| initLogManager | 初始化日志管理器 | 否 | 如果APP需要实现自定义的LogManager，则需要覆盖该方法。 |

#### UI基类

目的和用途：为所有UI提供基类支持。

技术标准：

* 1. 所有的业务APP的UI都必须从框架提供的UI基类继承。
  2. 基类需要自动处理：权限、身份、上下文、UI多语等。
  3. 页面基类只允许提供与展现相关和通用性的处理，不允许提供类似数据访问的能力
  4. 可以提供：日志、MVC视图展现、异常捕获、上下文访问、通用实体、字符及编码管理等能力，可以视情况进行扩展与精简。

##### BaseActivity

Activity基类，业务APP的Activity需集成自该基类。



子类可覆盖的方法：

| 方法名 | 方法作用 | 是否必须 | 补充说明 |
| --- | --- | --- | --- |
| initContentView | 初始化视图 | 是 | 初始化ContentView,请调用setContentView |
| initBase | 初始化基础组件 | 否 |  |
| initData | 初始化数据 | 否 |  |
| initMenu | 初始化菜单 | 否 |  |
| initMenuEvents | 初始化菜单事件 | 否 |  |
| initViews | 初始化视图 | 否 |  |
| initViewEvents | 初始化事件 | 否 |  |

##### BaseFragment

Fragment基类，业务APP的Fragment需集成自该基类。



子类可覆盖的方法：

| 方法名 | 方法作用 | 是否必须 | 补充说明 |
| --- | --- | --- | --- |
| onCreateView | 初始化视图 | 是 | 方法参数：  (BaseLayoutInflater baseLayoutInflater,  ViewGroup container,  Bundle savedInstanceState) |
| initData | 初始化数据 | 否 |  |
| initViews | 初始化视图 | 否 |  |
| initViewEvents | 初始化事件 | 否 |  |

##### BaseDialogFragment

DialogFragment基类，业务APP的DialogFragment需集成自该基类。



子类可覆盖的方法：

| 方法名 | 方法作用 | 是否必须 | 补充说明 |
| --- | --- | --- | --- |
| onCreateView | 初始化视图 | 是 | 方法参数：  (BaseLayoutInflater baseLayoutInflater,  ViewGroup container,  Bundle savedInstanceState) |
| initData | 初始化数据 | 否 |  |
| initViews | 初始化视图 | 否 |  |
| initViewEvents | 初始化事件 | 否 |  |

## ZTEAndroid-Async-Http

### 基本说明

异步通信组件，基于开源的android-async-http（<https://github.com/loopj/android-async-http>）的Http异步通讯框架修改封装。

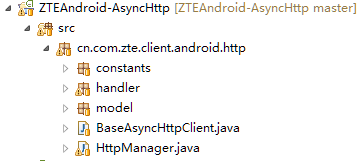


图3-5 Async-Http结构图

### API介绍

详情请参考API文档

## ZTEAndroid-Resource

### 基本说明

Android资源访问管理组件，可通过该组件访问系统资源。

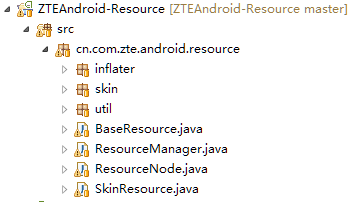


图3-3 Resource结构图

### API介绍

详情请参考API文档

## ZTEAndroid-ORM

### 基本说明

数据库访问组件，基于开源的ORMLite（<https://github.com/j256/ormlite-android>）的Java对象关系映射持久层框架修改封装。提供灵活的QueryBuilder来构建复杂的数据查询。DAO类能够非常方便地创建和删除数据库表。

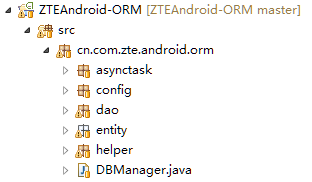


图3-4 ORM结构图

### API介绍

详情请参考API文档

## ZTEAndroid-EventBus

### 基本说明

事件总线组件，基于开源的EventBus（<https://github.com/greenrobot/EventBus>）框架修改封装。

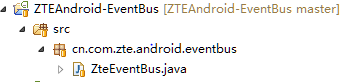


图3-6 Async-Http结构图

### 原理介绍

引入EventBus库，基于订阅-发布模型的事件总线机制。

EventBus是GreenRobot出品的Android系统的一个Event Bus类库，用来简化应用组件之间的通信。

当一个Android应用功能越来越多的时候，保证应用的各个部分之间高效的通信将变得越来越困难。为了这个问题，可以使用EventBus来进行**解耦**

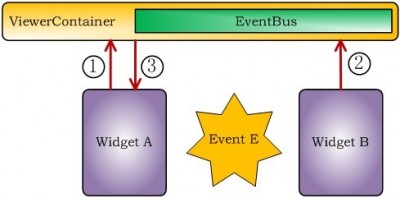
[](http://cms.csdnimg.cn/article/201305/21/519b0d442e3e7.jpg)

图3-6-1 EventBus原理图

#### 使用EventBus前的通讯模型

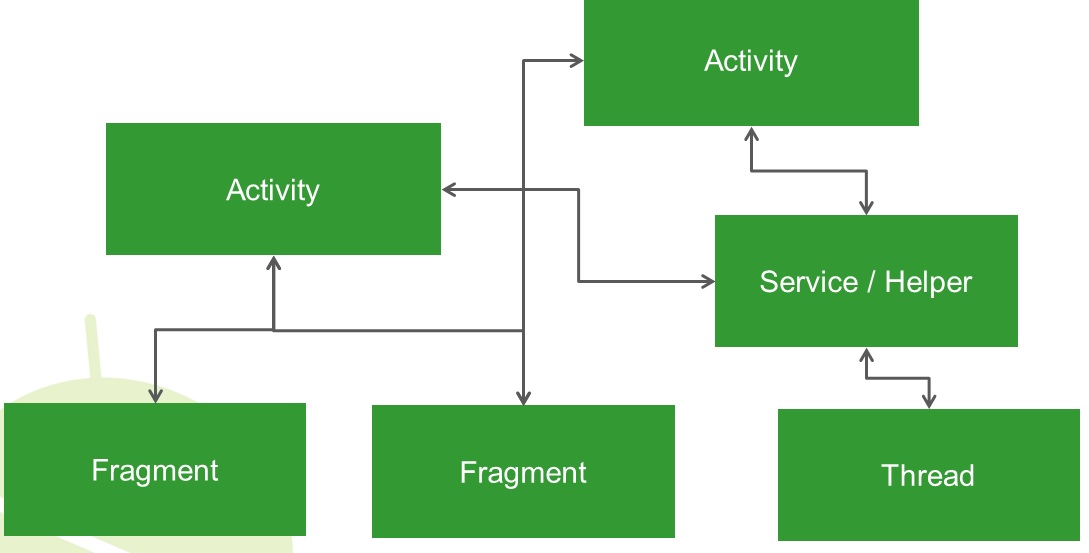


图3-6-2-1 EventBus模型图1

#### 使用EventBus后的通讯模型

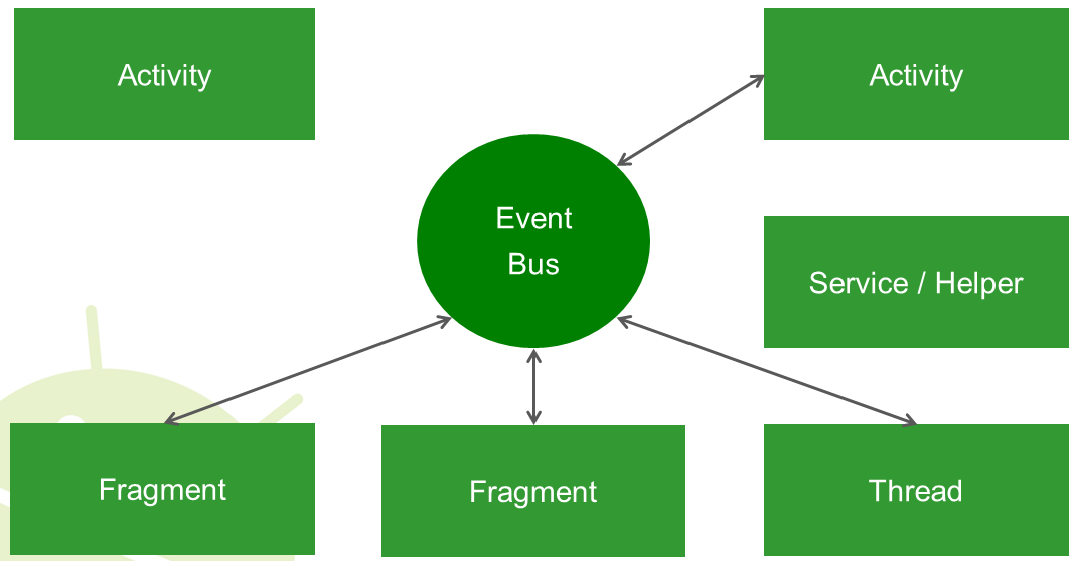


图3-6-2-1 EventBus模型图2

### API介绍

详情请参考API文档

## ZTEAndroid-LogMgr

### 基本说明

日志记录参考Log4j进行级别过滤，日志文件管理组件。

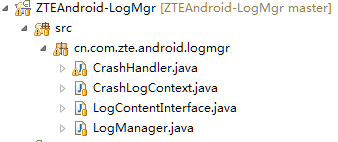


图3-7 LogMgr结构图

### API介绍

详情请参考API文档

## ZTEAndroid-Cache

### 基本说明

图片或文件的缓存管理组件。目前仅实现图片缓存。

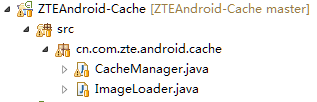


图3-8 Cache结构图

### API介绍

详情请参考API文档

// set the max number of icons (image width <= 50) to be cached in memory, default is 20

BitmapAjaxCallback.setIconCacheLimit(30);

// set the max number of images (image width > 50) to be cached in memory, default is 20

BitmapAjaxCallback.setCacheLimit(40);

// set the max size of an image to be cached in memory, default is 1600 pixels (ie. 400x400)

BitmapAjaxCallback.setPixelLimit(400 \* 400);

// set the max size of the memory cache, default is 1M pixels (4MB)

BitmapAjaxCallback.setMaxPixelLimit(2000000);

## ZTEAndroid-Widget

### 基本说明

可复用的自定义控件库,例如;日历控件,二维码扫描控件等等.

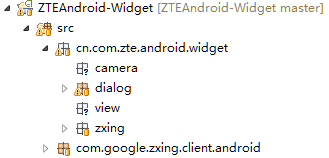


图3-9 Widget结构图

### API介绍

详情请参考API文档

## ZTEAndroid-AppUpdate

### 基本说明

应用升级组件,完成应用版本检测和自动升级组件。

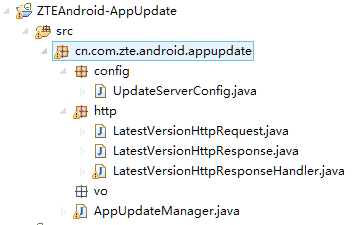


图3-10 AppUpdate结构图

### API介绍

详情请参考API文档

## ZTEAndroid-SecurityAuth

### 基本说明

安全认证组件, 完成用户身份认证、设备注册、单点登录。

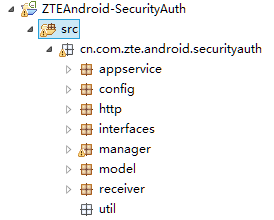


图3-11 SecurityAuth结构图

### API介绍

详情请参考API文档

## ZTEAndroid-Validation

### 基本说明

基于注解的表单验证组件,可提高表单验证的效率.

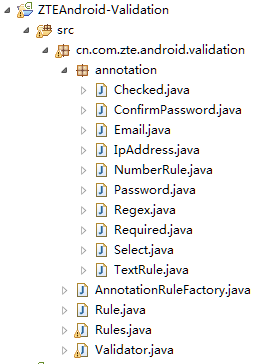


图3-11 Validation结构图

### API介绍

详情请参考API文档

#### 简单示例

一、通过注解声明校验规则:

/\*\* 用户账号输入框 \*/

@InjectView(R.id.*et\_qr\_user\_acc*)

@Email(order = 0, messageResId = R.string.*warn\_email\_error*)

**private** EditText etQrUserAcc;

/\*\* 用户姓名输入框 \*/

@InjectView(R.id.*et\_add\_user\_name*)

@Required(order = 2, messageResId = R.string.*warn\_nm\_error*)

@Regex(order = 3, pattern = "^[A-Za-z]+$", messageResId = R.string.*warn\_nm\_error*)

**private** EditText etUserName;

/\*\* 数据校验器 \*/

**private** Validator validator;

// 初始化validator

validator = **new** Validator(**this**);

二、需要校验时：

1. 校验单个视图

validator.validateView(etQrUserAcc);

1. 校验多个视图

List<View> lstListView = **new** ArrayList<View>();

lstListView.add(etQrUserAcc);

lstListView.add(etUserName);

validator.validateViews(lstListView);

1. 校验所有视图

Validator.validateAllView();

## ZTEAndroid-Common

### 基本说明

公共组件，包含一些公共的工具类。

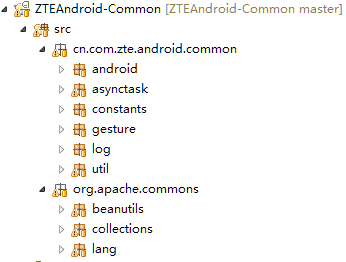


图3-2 Common结构图

### API介绍

详情请参考API文档

# 其他机制

## 依赖注入

### 原理介绍

引入Android平台下的依赖注入库RoboGuice（<https://github.com/roboguice/roboguice/>），ZTEAndroid-APP中的大部分主要组件例如BaseActivity、BaseFragment都已具备依赖注入能力。

简单的注入视图示例：

/\*\* 查询用户按钮 \*/

@InjectView(R.id.*btn\_qr\_user*)

**private** Button btnQrUser;

详细的注入能力请参考RoboGuice官方WIKI文档

（<https://github.com/roboguice/roboguice/wiki> ）。

## 换肤机制

### 原理介绍

使用拦截机制，拦截系统资源，替换为皮肤资源。



详情请参见：



# 编码规约

### Java通用编码规约

请参考公司java编码规范

### 资源目录和资源文件规约

先天限制：Android目前的系统限制，不能在资源文件夹下建立子目录。

Android开发过程中最容易失控的部分是资源文件的命名和维护。

#### drawable目录下只存放xml定义文件

drawable -- XML文件,包括selector、shape等文件

drawable-hdpi -- 对应的png图片 120像素/英寸屏幕

drawable-mdpi -- 对应的png图片 160像素/英寸屏幕

drawable-ldpi -- 对应的png图片 240像素/英寸屏幕

drawable-xdpi -- 对应的png图片 320像素/英寸屏幕

……

#### 所有资源文件名均为小写字符串，用下划线连接

格式:a\_b\_c\_d.xml

#### 布局文件中控件id命名

格式：采用匈牙利命名法

bt\_something is a Button

fr\_something is a Fragment

iv\_something is an ImageView

lv\_something is a ListView

tv\_something is a TextView

vp\_something is a ViewPager

ll\_something is a LinearLayout

rl\_something is a RelativeLayout

ib\_ something is a ImageButton

………

#### anim类型资源文件

格式：文件名以 **anim开头**.

例如: anim\_fade\_out.xml

#### drawable类型资源文件

格式：前缀\_<类型\_>名称<\_状态>

类型可选、状态可选

状态指激活和默认状态，例如：active和normal

前缀一般包括: selector、shape、bg、btn、ic等：

1、selector应该以selector前缀开始，跟随类型和名称，后缀为xml。

格式：selector\_<type\_>name.xml

例如: selector\_button\_home.xml;

2、shape应该以shape前缀开始，跟随类型和名称，后缀为xml。

格式：shape\_<type\_>name.xml

例如: shape\_button\_shape.xml;

3、背景应该以bg前缀开始，跟随名称和状态。

例如：

bg\_login\_panel.xml;bg\_login\_panel\_active.xml;bg\_login\_panel\_normal.xml;

4、按钮图片应该以btn前缀开始，跟随类型和名称。

例如：btn\_login.png;

5、图标应该以ic前缀开始，跟随类型和名称。

例如：ic\_menu.png;

#### layout类型资源文件

格式：类型\_名称

例如：

activity\_<activity\_name>.xml

fragment\_<fragment\_name>.xml

list\_item\_<category\_name>.xml

dialog\_<category\_name>.xml

popwindow\_<category\_name>.xml

map\_<category\_name>.xml

#### menu类型资源文件

格式：menu\_<category\_name>.xml

#### style类型资源文件

格式：style\_<category\_name>.xml

#### string类型资源文件

格式：string\_<category\_name>.xml (默认文件除外)

格式：array\_<category\_name>.xml (默认文件除外)

### 中文处理规约

#### 所有中文均在资源文件中定义

避免在java源码中硬编码中文

#### 建立中文错误信息专用string资源文件

/res/values/error\_message.xml