Android架构篇

工程师 李洪江

1. 什么是组件化开发

组件化开发，也叫AAR开发。

组件化开发其中难点是业务组件化。

1. 为什么要组件化？

Android架构从最早的按功能分包，到后来的按业务模块分包，单一工程架构。现在主流是组件化开发和插件化开发。随着项目功能，业务越来越多，项目耦合度增强。人员更替，项目维护、学习成本高。每改一个功能，app回归测试重复性增大。改一小部分，导致整个项目重构，花费时间太多，效率低下。

组件化优势：

1. 新人学习项目成本低，代码耦合度下降，方便功能拆卸。
2. 加快开发和编译效率。
3. 便于每个模块版本管理。
4. 便于开发人员为每个模块写单元测试。
5. 只是修改某个业务模块功能，测试人员不需要回归测试其他功能。
6. 云宠APP组件化实践

云宠APP四层架构：



每层含下列库：



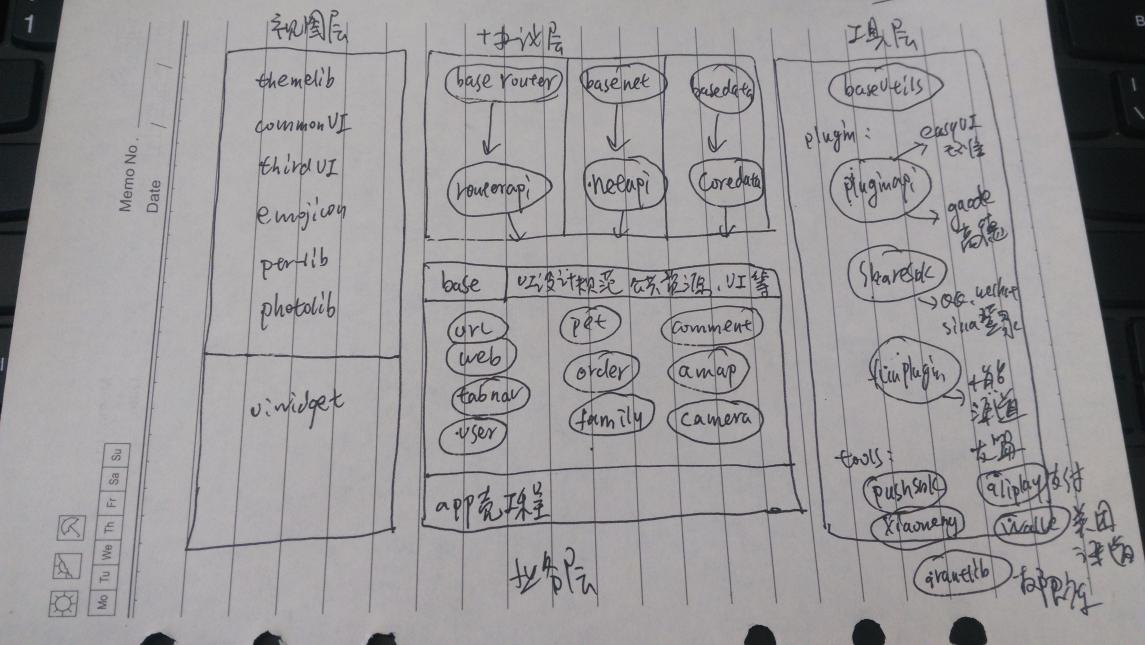
项目依赖：



设计原则：

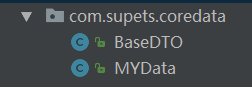
1. 协议层尽量不要和UI层有关系。
2. 业务层是通过协议层通信的。
3. 业务层之间不能相互依赖。
4. 工具服务于其他层。

依赖关系图：



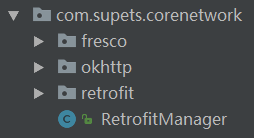
1. 基础库设计

1基础数据库basedata



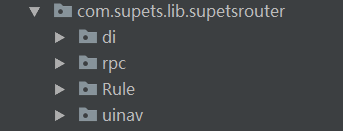
定义数据传输协议：序列化

2 基础网络库basenet



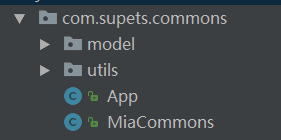
定义基础网络架构：Rest风格

1. 基础通信库baserouter



定义了业务UI通信模式：

1. 静态注册，使用原生Intent方式。
2. 动态注册接口，采用RPC方式。
3. 隐式Intent启动，采用rest风格注解框架
4. 基础工具库 baseutils



主要作用：

1. App上下文管理
2. App 协议解析工具
3. 时间格式化工具
4. 系统相关工具
5. UI相关工具
6. 技术使用
7. 网络技术

Volley官方网络通信库。

底层技术SDK<9采用apache httpClient。SDK>=9采用了HttpConnection实现。

在SDK23 已经阉割，httpclient实现库被独立出来，需要使用的话单独使用。

原理参考:

<https://github.com/android-cn/android-open-project-analysis>

Volley 的特点：特别适合数据量小，通信频繁的网络操作。

Volley大多基于接口设计，可配置性强。常见支持https，图片缓存实现，zip解压缩，数据加密等使用。

Retrofit Square出品的，是一个采用适配器架构和注解配置的RESTful风格设计网络库。

底层是依赖OkHttp实现的。

简洁好用，搭载了拦截器，方便实现日志输出，统一请求参数配合等。

1. 缓存技术

GreenDao，Ormlite，RealIm

文件存储:适合一些配置类的信息，比如版本升级，渠道信息。

数据库存储：适合一些数据量大的，比如收货地址信息。

内存存储：适合一些频繁操作的数据。比如登录用户信息。

1. 数据池技术

在数据改变，影响其他地方数据一致性情况下，又不需要永久存储的情景下。

数据同步缓存到一个池子，再从池子拿出来。

比如，社交应用的赞，评论数，评论等同步更新。

1. 运行时注解和编译时注解技术

编译时注解：butterknife，dragger

运行时注解：retrofit，EventBus，自定义URI框架

1. AOP技术运用

常见写法：

1经典的基于代理的AOP

2 @AspectJ注解驱动的切面   
3 纯POJO切面   
4 注入式AspectJ切面

比如，统计方法执行时间，android behavior使用，一些操作需要登录再能继续。

1. 注解使用

Java

@Override @Deprecated @SuppressWarnings

Android

Nullness注解、资源注解、线程注解、值约束注解、权限注解、返回值注解、CallSuper注解、Typedef 注解、代码可访问性注解等。

自带注解很重要。

比如:方法过时，资源标识类型，限定输入范围等。

1. 泛型的使用

在封装类，方法等时候。

常见的findViewById方法的封装。这个在SDK 26 已经不需要，里面已经泛型化。

在封装baseAdapter时候。

1. 代理的使用

在android方面，事件的回调可以叫代理。

静态代理使用。常见写法。

动态代理使用。

1. 反射的使用

反射是很多框架基础。

最早EventBus实现是基于反射。

运用：反射同步数据，反射获取类名。

1. 图片加载

Glide，Volley，Picasso，Fresco

附录1：常用开发工具助手

1 fir插件

2 nexus仓库

3 jenkins集成

4 markdown插件

5 git插件

6 alibaba开发规范插件

7 android layout ID Converter插件

8 形状代码生成器 <http://shapes.softartstudio.com/>

附录2：android studio快捷键

大小写转换快捷键ctrl+shift+U

提取全局变量提取全局变量:Ctrl+Alt+Full

提取全局变量提取局部变量：Ctrl+Alt+Val

提取全局变量提取方法：Shit+Alt+Method

复制块：Ctrl+D

删除：Ctrl+X

修改名称: Ctrl+Shift+R

上下移动代码 : Alt + Shift + Up/Down

注释代码(//) Ctrl + /

注释代码(/\*\*/) Ctrl + Shift + /

格式化代码 Ctrl + Alt + L

清除无效包引用 Alt + Ctrl + O

查找 Ctrl + F

查找+替换 Ctrl + R

删除行 Ctrl + Y

扩大缩小选中范围 Ctrl + W/Ctrl + Shift + W

快捷生成结构体 Ctrl + Alt + T

快捷覆写方法 Ctrl + O

快捷定位到行首/尾 Ctrl + Left/Right

折叠展开代码块 Ctrl + Plus/Minus

折叠展开全部代码块 Ctrl + Shift + Plus,Minus

文件方法结构 Ctrl + F12

查找调用的位置 Ctrl + Alt + H

大小写转换 Ctrl + Shift + U