# Android客户端架构设计

## 客户端架构设计

**客户端设计** 目的是整体设计客户端App，架构上打好铺垫.

### Android客户端架构设计

主要从以下几个方面进行设计：**MVP设计风格**、**整体架构**、**日志系统**、**网络系统**、**本地存储**、**Test模块**.

#### MVP设计风格

MVP即“Model —— Presenter —— View”，应用在Android中可以实现Activity和业务逻辑的解耦，简化Activity的规模。现在Gitbub上暂时没有合适的通用的MVP框架，我们可以基于[Gitbub: Android MVP Demo](https://github.com/antoniolg/androidmvp) 开源项目实现MVP模式。

基本原理：

View：接口，声明所有的View相关的操作，包括GetValue、SetValue、Progress、Navigator等。MVP思想的核心，通过提取View接口，实现了Activity和Presenter的解耦。

Activity： View接口的实现，初始化界面，初始化View中的控件，调用Presenter完成业务逻辑

Model：涉及到的数据对象，以及对数据的操作，比如可以把IO操作放到Model层

Presenter：业务逻辑的实现，业务逻辑处理完成之后通过View接口操作UI，作用的数据是Model.

#### 整体架构

依赖注入型框架：

-- **Dagger**

**阿里手淘项目推荐使用**  
项目地址：[https://github.com/square/dagger](https://github.com/square/dagger" \t "http://www.jianshu.com/p/_blank)  
原理剖析文档：[http://square.github.io/dagger/](http://square.github.io/dagger/" \t "http://www.jianshu.com/p/_blank)

-- **ButterKnife**  
**视图注入型框架**  
项目地址：[https://github.com/JakeWharton/butterknife](https://github.com/JakeWharton/butterknife" \t "http://www.jianshu.com/p/_blank)  
文档介绍：[http://jakewharton.github.io/butterknife/](http://jakewharton.github.io/butterknife/" \t "http://www.jianshu.com/p/_blank)

Butterknife的具体用法参考文档说明，这里简单介绍其特性。

支持Activity的View注入，比如简化Activity中的findViewById()

支持View中的View注入，比如简化View.findViewById()

支持View的事件回调

注入示例代码如下,更多注入示例请参考文档：

public class ExampleActivity extends Activity {

@Bind(R.id.title) TextView title;

@Bind(R.id.subtitle) TextView subtitle;

@Bind(R.id.footer) TextView footer;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.simple\_activity);

ButterKnife.bind(this);

// TODO Use fields...

}

@OnClick(R.id.submit)

public void submit(View view) {

// TODO submit data to server...

}

@OnClick(R.id.submit)

public void sayHi(Button button) {

button.setText("Hello!");

}

}

#### 日志系统

日志系统要完成以下几个功能：

Debug时的日志需求

日志写入文件的需求，方便上线查看

日志上报

日志加密

系统日志监控：日志奔溃信息监控

可以修改Github开源项目[格式化日志输出项目： Logger](https://github.com/orhanobut/logger" \t "http://www.jianshu.com/p/_blank) 和 [日志收集项目：Log](https://github.com/licong/log" \t "http://www.jianshu.com/p/_blank) 达到要求。

#### 网络系统

[Gitbub: OkHttp](https://github.com/square/okhttp" \t "http://www.jianshu.com/p/_blank)

OkHttp封装：HttpGet、HttpPost、File Upload，OKHttp的使用要求创建Request，填写RequestBody。参照医行者Messaage的封装形式，将创建Request的过程进行封装，由最简单的请求信息和回调接口构成一个Message，并对Message进行统一处理。

#### 本地存储

本地存储要解决的问题：

数据库存储

数据库的版本管理：版本升级、数据迁移等

第三方框架：

-- **GreenDAO**

Android Sqlite orm 的 db 工具类

项目地址：[https://github.com/greenrobot/greenDAO](https://github.com/greenrobot/greenDAO" \t "http://www.jianshu.com/p/_blank)

文档介绍：[http://greendao-orm.com/documentation/](http://greendao-orm.com/documentation/" \t "http://www.jianshu.com/p/_blank)

官网网址：[http://greendao-orm.com/](http://greendao-orm.com/" \t "http://www.jianshu.com/p/_blank)

**GreenDao使用方式**

创建 "Green Dao Generation Project" 来生成 entities and core classes( DaoMaster, DaoSession .etc);

在Android Project中使用生成的Dao，

GreenDao生成：

public class ExampleDaoGenerator {

public static void main(String[] args) throws Exception {、

// 这两个参数是：数据库版本号和schema名称

Schema schema = new Schema(1000, "de.greenrobot.daoexample");

addNote(schema);

//addCustomerOrder(schema);

// 生成类，同时可以指定java文件位置

new DaoGenerator().generateAll(schema, "../DaoExample/src/main/java");

}

private static void addNote(Schema schema) {

Entity note = schema.addEntity("Note");

note.addIdProperty();

note.addStringProperty("text").notNull();

note.addStringProperty("comment");

note.addDateProperty("date");

}

}

针对版本升级的问题，GreenDao也有合适的解决方式：

数据库版本升级问题：

/\*\* WARNING: Drops all table on Upgrade! Use only during development. \*/

public static class DevOpenHelper extends OpenHelper {

public DevOpenHelper(Context context, String name, CursorFactory factory) {

super(context, name, factory);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

Log.i("greenDAO", "Upgrading schema from version " + oldVersion + " to " + newVersion + " by dropping all tables");

//dropAllTables(db, true);

//onCreate(db);

// to do update database

}

#### Test模块

**Android单元测试**

-- **robolectric**

项目地址：[https://github.com/robolectric/robolectric](https://github.com/robolectric/robolectric" \t "http://www.jianshu.com/p/_blank)

Demo 地址：[https://github.com/robolectric/robolectricsample](https://github.com/robolectric/robolectricsample" \t "http://www.jianshu.com/p/_blank)

文档介绍：[http://robolectric.org/](http://robolectric.org/" \t "http://www.jianshu.com/p/_blank)

特点：(1). 不需要模拟器在一般 JVM 就可以运行测试用例(该框架在Java层实现了一套接口模拟Android，所以可以直接在JVM上跑Android Unit Test)

(2). 能完成在真机上的大部分测试包括感应器

其他的测试用例及相关模块 Mock 可见：android-mock, mockito, easy-mock

**robolectric**

举个例子:

Andorid中的代码

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

TextView textView =(TextView)findViewById(R.id.textView1);

textView.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(MainActivity.this, SecondActivity.class));

}

});

}

}

对应的测试类，MainActivityTest的代码：

@RunWith(RobolectricGradleTestRunner.class)

@Config(constants = BuildConfig.class, sdk = 21)

public class MainActivityTest {

@Test

public void testMainActivity() {

MainActivity mainActivity = Robolectric.setupActivity(MainActivity.class);

mainActivity.findViewById(R.id.textView1).performClick();

Intent expectedIntent = new Intent(mainActivity, SecondActivity.class);

ShadowActivity shadowActivity = Shadows.shadowOf(mainActivity);

Intent actualIntent = shadowActivity.getNextStartedActivity();

Assert.assertEquals(expectedIntent, actualIntent);

}

}

-- **Monkey**

Monkey是Android中的一个命令行工具，可以运行在模拟器里或实际设备中，基于adb shell命令还控制程序包进行测试。它向系统发送伪随机的用户事件流(如按键输入、触摸屏输入、手势输入等)，实现对正在开发的应用程序进行压力测试。Monkey测试是一种为了测试软件的稳定性、健壮性的快速有效的方法。Monkey测试对象是应用程序包。

Monkey测试的步骤：  
1、#adb shell  
2、#monkey -p com.android.calculator2 -v 500 （-p packageName -v LogLevel 500即伪随机事件个数）  
更多地命令组合和参数可以查看：  
Google Monkey官网：[http://developer.android.com/tools/help/monkey.html](http://developer.android.com/tools/help/monkey.html" \t "http://www.jianshu.com/p/_blank)  
Android-Monkey-Test：[http://ihongqiqu.com/2015/12/24/Android-Monkey-Test/](http://ihongqiqu.com/2015/12/24/Android-Monkey-Test/" \t "http://www.jianshu.com/p/_blank)

**第三方测试工具**

-- **Tencent APT**

APT 是腾讯开源的一个 Android 平台高效性能测试组件，提供丰富实用的功能，适用于开发自测、定位性能瓶颈；测试人员完成性能基准测试、竞品对比测试