**前言**

      相信大家在开发Android应用程序的过程中都会遇到很多重复的功能，许多基础的功能几乎在我们开发过程中无时不刻要接触到。这个时候很多人都会选择依靠Afinal，XUtil，thinkAndroid等一些第三方开发框架。通过这些框架，我们就不再需要一个个的写异步线程去实现网络请求。不再需要自己写图片缓存操作。只需要简单的几句话。就能够快速实现我们想要的功能。相比以往一个个功能自己几十行几百行代码的实现。这些快速开发能够帮助我们提高不少的效率

      然而，随着时间的推进。Android的版本每年一个版本的迭代。许多的开发要求也慢慢在变化。但是这些第三方快速开发框架在没有及时更新的情况下。很容易就会因为没有替换相应的底层库导致问题百出。拿http来说。很多框架还用着google已经不再提倡在2.3以上系统使用的HttpClient。因此，选择一个能够及时更新迭代的库很重要。

      不过，对于多数公司来说。往往是自己搭建一套同样的库。如果你有兴趣，从github下载这些第三方快速开发框架的源码下来研究。就会发现，其实这些库的原理都大同小异。很多都是选择一些目前github上比较火的库进行二次封装，有的则是根据需求。自己封装好相应的实现功能类。通过开放一致的接口调用。这样，开发者们只需要简单的几句调用。剩余的工作都是框架内部封装来实现。

      相信只要仔细研究透这些库的原理。自己也能够试着搭建出一个符合自己需要的快速开发库。本系列教程，我就跟大家介绍介绍怎么搭建出一个类似的库。

**原则**

      我很赞同我一个领导说的话。设计是十分重要的，设计必须要大于需求。因此，我们首先要考虑好，怎么搭建好这个框架。这个框架需要提供哪些内容，并不是简单的提供一个http get请求的功能就行了。很多童鞋往往需要这个库能够支持post请求。自定义Header等需求。

      想要搭建一个结构良好。便于维护。并且简单易用的快速开发框架需要遵守以下几个原则：

1. **合理选择需要的模块：**并不是越多模块越好，需要考虑好哪些是大家都需要的。过多的内容反而会使这个框架过于臃肿，包的大小太大。这一点往往是开发者选择时候最看中东西。使用你的库。结果自己的APK需要增加好几M。肯定不合适。因此我们要做的是在精不在多。只要提供好一些基础常用，效率高。使用方便的库就行了
2. **做好二次封装，方便迭代替换：**就算再好的库，也有过时的一天。所以不能想着你用的是目前网上公认效率最高。速度最快的库。就直接拿来用。世态无常。当这个库不再受到大家推崇的时候，你要做迭代替换成其他的库。这个时候麻烦就大了。因此。你需要注意。不管你用了什么库。都要给他做一次封装。提供统一的入口。特别要注意参数这一块。举个栗子，一些http请求模块，往往会把参数封装成自己自定义的map来传递，这时候我们就要通过自己的封装，把原本第三方自己定义的参数以系统常见参数的方式来替代。这样就算你实现库更换了。用户也只需要更新一下你的库就行了。不需要再根据你替换的库。所有用到的自定义参数类型都改一遍。这样的迭代成本太高！
3. **接口设计，多样丰富，简单易用：**我们要设计好相关模块在这一需求上尽可能多的场景，例如，http请求。很多开发者会需要自定义header数据。甚至cookie机制。这些我们都是需要考虑进去。并且在设计接口的时候。要尽可能的用通俗易懂的命名，让开发者不需要看注释就能够理解你的接口怎么使用是最高的境界。接口的设计要尽可能提供给开发者自由度，例如参数，直接提供一个map。用户想要怎么封装map都可以。也可以学习volley的接口化编程，让开发者有特殊需求的时候。直接实现你已经定义好的接口来完成自己想要做的事情。不需要再去改动你的源码。
4. **UI归UI，逻辑实现归逻辑实现：**很多人喜欢在框架里除了一些基本功能模块之外封装很多的ui组件，其实我并不太赞同。我更建议一些实现逻辑的库集合一个框架。ui库集合一个框架。因为并不是所有的ui他们都会用到。ui的库往往又会带着很多资源文件。这样很大程度上会导致框架占用的资源过多。
5. **预判错误：**当我们设计接口开放的时候，一定要记得考虑全面。预判这个接口可能会有什么样的异常情况。模拟出不同场景。可能是开发者传递过来的参数会是空的。可能这个接口在调用的时候需要网络。没网络的情况下要怎么返回数据告知开发者是什么原因。这个是十分重要的。如果从框架这边不能够告知开发者错误原因所在。这会加大开发者排查错误的难度。
6. **版本兼容：**虽然说接口设计一定要尽可能避免版本迭代需要做大改动。但是难免会有写情况会遇到。因此我们需要考虑通过对框架版本做相应管理判断来兼容各个版本。通过对不同版本的判断，处理相应的逻辑。保证就算你框架升级了。旧版的框架依然能够为用户所用。
7. **提供生命周期：**对于一些复杂流程的模块设计，特别是异步操作类型的模块。一定要提供清晰的声明周期供开发者调用重写。例如异步开始，异步进度，异步结束，异步异常等。这样的话在这些操作下。开发者可以根据自己的需要在不同的声明周期阶段来实现一些ui效果展示给用户。提高框架的扩展性。

**总结**

      以上就是个人总结出的搭建一个快速开发框架需要遵守的设计准则。可能不够全面，也希望有补充的朋友可以多多建议。在了解这些原则之后，我们就可以开工，开始一步步来搭建适合我们自己的快速开发框架。接下去，我会通过对每个常用模块的各个流行的库进行优势分析。并且提供简单的设计思路，我们一起探索如何选择适合自己的实现库，并封装出适合各个库的开放接口！

Android快速开发框架搭建之第三方库选择-Http篇（二）

* Android
* 快速开发
* 框架，搭建

      在了解了搭建框架的原则之后。我们就一步步从常用的几个功能模块来开始搭建。

      首先，也是最重要，最经常使用的一个模块:Http请求。几乎可以说除了离线应用和游戏。90%以上的app都需要请求服务器来获取数据进行展示。Http是我们开发应用接触多的一个功能模块。从使用频率以及重要性来说，它都是优先级最高的一个操作。因此，我们需要选择一个效率足够高，并且能够管理的好的http请求库来使用。

**选择**

      这里，我们列举出目前github最为出色的几个库，提供给大家选择。当然，如果大家有更好的库推荐，也可以提供给楼主补充。

      1.Volley

            Volley是google官方在2013年Google I/O大会开源的一个Http请求库。其集合了http请求与网络图片加载缓存的功能于一身。成为google大力推荐使用的库。并且，它能够自动根据所运行的系统版本。自动选择使用HttpClient还是HttpUrlConnection。并且由于Volley的结构是使用接口话编程的框架。使其具有十分强大的扩展性。任何你想要修改的地方都可以使用接口继承的方式来自定义实现。Volley还针对网络及缓存做了相关的队列管理，能够很好的对其中的线程进行优先级排序以及取消等操作。

            下载地址：<https://android.googlesource.com/platform/frameworks/volley>

      2.AsyncHttpClient

            AsyncHttpClient是github使用量很大的一个http请求库。其封装了强大的异步线程池来实现http请求。并且提供了大量常用的使用方式。特别是实现了文件上传下载的功能。只需要简单的几句调用就能够实现以往复杂的文件下载上传操作。不过由于其底层实现使用的是HttpClient。目前Google已经不推荐使用HttpClient在2.3以上版本使用。在5.1源码之后还删除了HttpClient部分的源码。因此如果你要使用AsyncHttpClient。需要注意将引用的部分使用Apache引入的jar库来使用。

            下载地址：<https://github.com/loopj/android-async-http>

      3.OKHttp

            OKHttp是国外知名的科技公司Square出品的开源http请求库。Square在开源界是神一般的存在，他们为我们提供了太多强大的库。甚至很多库都被google集成到源码中去使用。OKHttp便是其中一个，在新版本的android源码中，已经使用OKHttp替代了HttpUrlConnection的实现，它提供了与HttpUrlConnection一样的功能和api。能够让我们快速的上手使用。并且从请求效率上来说。OKHttp是无人能匹敌的。这也是Google选择它集成到源码中替代原有实现的原因。

            下载地址：<https://github.com/square/okhttp>

      那么。这么多的库，选择哪一个更合适呢。这个就需要大家自己根据自己的需求去选择了。在了解过各个库的优缺点之后。大家可以任意选择一个库进行封装。这里，我个人推荐一种组合方案。使用Volley+OKHttp。Volley来实现队列管理调度。OKHttp来实现http的实现部分。这样，利用两者的优点结合，相信能够有一个很好的效果。

**封装**

不管你选择的是哪一个库。最重要的就是要封装好一个统一的接口。这样可以避免你在替换底层实现库的时候因为更换的库实现api差距很大而需要做大量的改动。

      这里，我提供一个案例：

public void doPost(final Context mActivity,String url, final Map<String,String> param, final OnHttpListenner mListner, final boolean showProgress){

StringRequest stringRequest = new StringRequest(Request.Method.POST,url,new Response.Listener<String>(){

@Override

public void onResponse(String s){

try{

BaseRequestEntity result=JSON.parseObject(s, BaseRequestEntity.class);

try {

mListner.OnSuccess(result);

} catch (PackageManager.NameNotFoundException e) {

e.printStackTrace();

}

}catch (Exception ex){

ToastUtil.showShort(mActivity,"请求异常，请稍后重试");

}

}

},new Response.ErrorListener(){

@Override

public void onErrorResponse(VolleyError volleyError) {

mListner.OnError();

}

}){

@Override

protected Map<String, String> getParams() throws AuthFailureError {

param.put("venue","1");

return param;

}

};

stringRequest.setRetryPolicy(new DefaultRetryPolicy(

60000,

1, 1.0f));

mRequestQueue.add(stringRequest);

}

在这个方法里默认使用的是post方式请求。参数封装为map传递进来。接入到Volley的getParams中。提供Context，url，是否显示加载框，最重要的是，通过OnHttpListenner回调来渗透到http请求的各个声明周期。方便外部做相关阶段的自定义处理。

### 开发android app，会用到哪些开源框架？

* Android
* 开源框架

开发android app，都经常会用到哪些开源框架呢？大家能不能介绍下自己常用的~

0 4

## 共收到4条回复

[浩海如烟](http://my.jikexueyuan.com/9160833719) 41天前[#1楼](http://qun.jikexueyuan.com/android/topic/113#reply1)

**UI层**

图片加载:之前用的Picasso，非常方便。但因为picasso是自带的请求处理，没办法和已有的 http 请求统一调度。

依赖注入：Butterknife ，小而全的依赖注入库。

**数据层**

Http/REST 请求：如果是严谨的REST API , Retrofit 会非常好用！各种 annotation 用起来省事又省心！唯独不好的地方是很多时候我们的接口不是完全的 REST 结构，同时需要针对每个 request 做单独的处理，这时候 Retrofit 就显得有些尴尬了。这时候我会用 volley，volley 的请求自定义和管理支持很好。

异步网络请求处理：是否曾经被网络请求的各种异常情况弄得焦头烂额？直到用到了 RoboSpice，事情变得简单许多。对网络请求的再次封装，将请求的各种结果回调变得简便，同时对 Retrofit , Okhttp,Ormlite 等支持都非常完善了，有了它，生活更 easy 了。

JSON解析：最通用的是 Gson ，用起来也很方便，但从效率上来讲 Gson 不是最快的， FastJson 和 Jackson 各有优势

数据库：ormLite和greenDao都挺好的。

**其他**

消息事件队列：以前用 Guava ，后来发现有针对 Android 平台设计的 Otto 性能更优

0[评论](http://qun.jikexueyuan.com/android/topic/113#editorBox)

[爱搞siao](http://my.jikexueyuan.com/4214544749) 41天前[#2楼](http://qun.jikexueyuan.com/android/topic/113#reply1)

图片加载，UIL或者Picasso；

数据库，ormLite或者greenDao；

网络层，apcahe的http-common或者square的okhttp；

聊天，XMPP；

JSON解析，fastJson；

动画，NineOldAndroids。

0[评论](http://qun.jikexueyuan.com/android/topic/113#editorBox)

[鹰之水舞](http://my.jikexueyuan.com/5410100773) 41天前[#3楼](http://qun.jikexueyuan.com/android/topic/113#reply1)

好用且被广泛使用：

1.universal-imageloader

2.okhttp/android-async-http

3.greendao

4.acra

5.appirater

6.atinject

7.disklrucache

8.太多了

0[评论](http://qun.jikexueyuan.com/android/topic/113#editorBox)

[Yan小坏](http://my.jikexueyuan.com/xiaohuai) 10天前[#4楼](http://qun.jikexueyuan.com/android/topic/113#reply1)

我用到的：

1.UI  图片加载  ImageLoader

2.数据库及http   volly 或者 afinal ，但是数据库升级部分需要自己写，所以我数据库部分没有用他们的，自己封装了一层，方便数据库升级数据迁移。

3.下拉刷新组件  pull-to-refresh-listview

其余的我以这个网址代替：

<http://blog.csdn.net/qq530918474/article/details/38920265>

建议：

框架固然方便，但尽量看下框架里边的思路，能看懂多少看多少，常自己DIY封装一些东西发到git上，这样对开发学习很有帮助。