本次主要针对之前用过的几种网络请求进行对比，以此记录  
****一，android-async-http****  
由于android6.0以后不再支持HttpClient，所以类似于android-async-http这类基于httpClient的建议废弃，不再使用  
****二，Volley(轻量级网络交互框架)****  
集成了androidAsyncHttp和ImageLoader框架特点，android2.2以及以下版本默认使用HttpClient，android2.3以及以上版本使用HttpUrlConnection（）  
优点：适用于数据量小，通信频繁，扩展性较强。  
缺点：对于大数据量的操作，以及下载文件，上传文件 Volley不是很适合，并且不支持同步请求，局限了开发模式  
****三，OKHttp（建议使用）Square出品****  
android4.4开始HttpUrlConnection的底层实现采用的是OKhttp。OkHttp支持android2.3以上版本 java最低要求1.7  
优点：  
1,高性能HTTP库，支持同步，异步请求，支持文件上传下载很高效，  
2,支持spdy，http2，共享同一个socket来处理同一个服务器的所有请求，如果SPDY不可用，特通过连接池来减少请求延时  
3.无缝支持GZIP来减少数据流量  
4.缓存响应数据减少重复的网络请求  
5.自动处理常见的网络问题，支持自动重连，例如二次链接，SSL握手问题  
缺点：  
请求后的回调方法 不在主线程无法直接更新UI 需要自己通过handler或者其他方式去更新ui  
****四，Retrofit2.0 Square出品****  
底层网络请求采用OkHttp，采用注解的方式来制定请求方式和url地址  
如果是标准的RESTFUL API，那么采用Retrofit2.0代码里会减少很多。建议可以结合OKHttp使用来支持一些非标准的HTTP方法  
如果程序中集成了okhttp，Retrofit默认会使用OkHttp处理网络层请求，建议结合OKHttp，Rxjava使用

注：  
volley, retrofit, android-async-http 帮你封装了具体的请求，线程切换以及数据转换。  
而OkHttp 是基于http协议封装的一套请求客户端，虽然它也可以开线程，但根本上它更偏向真正的请求，跟HttpClient, HttpUrlConnection的职责是一样的。