（一）HttpClient和HttpURLConnection的区别

Android主要提供了两种方式进行网络请求：HttpClient与HttpUrlConnection。这两种方式都支持https协议，以流的形式进行上传和下载、配置超时事件、IPV6、以及连接池等功能。

HttpClient拥有众多API，实现较稳定，bug数量少。但难以扩展，维护成本高。Android6.0后被移除。

HttpUrlConnection提供简单、轻量级API，易于扩展。但Android2.2前有个重大bug。2.3后修改了bug并且提供了压缩和缓存机制，有效提升了网络性能。

（二）Volley，OkHttp，Retrofit（三大常用Android网络框架）之间的区别和核心原理和使用场景

1、Volley是Google推出的网络通信框架，适合数据量小但通信频繁的网络操作。但不适合大文件下载。

特点是：（1）可进行Post、Get网络请求与图像异步处理请求（2）对网络请求进行排序与优先级处理（3）对网络请求进行缓存（4）多级别取消请求（5）与Activity生命周期联动

它的工作原理是先将请求加入缓存队列，并通过cacheDispatcher查询本地是否缓存本次请求结果，如果命中，则从缓存中解析结果并返回主线程；如果没有命中则将请求添加到网络队列，并通过networkDispatcher发送网络请求，获得并解析响应结果，将结果写入缓存并返回主线程。

2、OkHttp是Square团队开发的支持Http2/SPDY的网络通信框架，即支持共享同一个Socket处理同一个服务器的所有请求，若SPDY不可用，则通过连接池来减少请求延时。支持重连机制、缓存响应数据及GZIP减少数据流量。

OkHttp核心设计模式是拦截器责任链模式，采用责任链的模式来使每个功能分开，每个拦截器自行完成自己的任务，并且将不属于自己的任务交给下一个，简化了各自的责任和逻辑，实现了网络请求。这样设计的好处在于使每个责任链可以分层实现缓存、压缩、网络IO和请求等功能，并且可以对响应的数据做其他的逻辑处理。

OkHttp适用于数据量大的重量级网络请求。

3、Retrofit是基于RESTful风格推出的网络请求框架封装，是基于OKHttp的网络请求框架的二次封装。其底层是通过OKHttp进行网络请求，而Retrofit仅负责网络请求接口的封装，从而简化了用户网络请求的参数配置，还能与Rxjava结合。

Retrofit采用了大量设计模式封装OkHttp，它的核心工作原理是将Http请求抽象为Java接口，在接口中用注解描述和配置网络请求参数。Retrofit使用动态代理的方式，动态地将网络请求接口的注解解析成HTTP请求。最终通过OKHttp执行Http请求。

Retrofit在任何场景下都优先选择，尤其是后台Api遵循RESTful风格且项目中使用RxJava的场景。

（三）网络请求缓存处理，okhttp如何处理网络缓存的？

Http网络请求缓存处理：

强制缓存：当用户端第一次请求数据是，服务端返回了缓存的过期时间（Expires与Cache-Control），没有过期即可以继续使用缓存；如果过期则不使用缓存，无需再请求服务端。

对比缓存：当用户端第一次请求数据时，服务端会将缓存标识（Etag/If-None-Match与Last-Modified/If-Modified-Since）与数据一起返回给用户端，用户端将两者都备份到缓存中 ，再次请求数据时，用户端将上次备份的缓存

标识发送给服务端，服务端根据缓存标识进行判断缓存是否过期，假如返回304，则表示缓存可用，从缓存中读取数据；假如返回200，标识缓存不可用，请求服务端，并使用最新返回的数据。