

## OkHttp源码解析

### 五大拦截器分析

- 重试与重定向拦截器, 桥接拦截器
- http缓存机制与缓存拦截器
- 连接拦截器与请求拦截器

往期视频咨询阿媛老师QQ:2807762965



享学课堂,让学习成为一种享受!

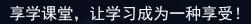


#### Lance

某游戏公司主程,前爱奇艺高级工程师。多年移动平台开发经验,涉猎广泛,热爱技术与研究。主要对NDK、架构与性能优化拥有深入的理解及开发经验。授课严谨负责。

• QQ: 3265104286



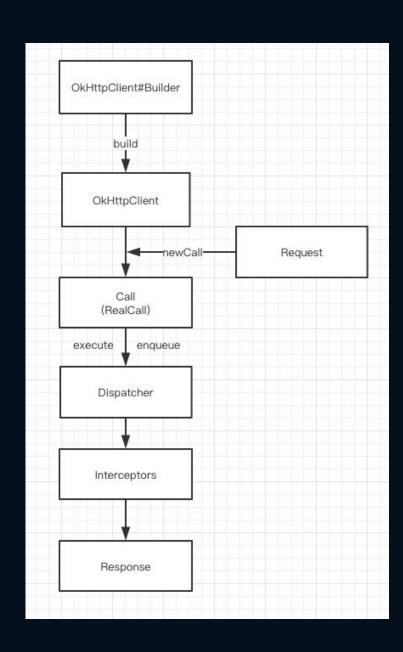




关注享学课堂 免费领取课程

#### ▮上期回顾





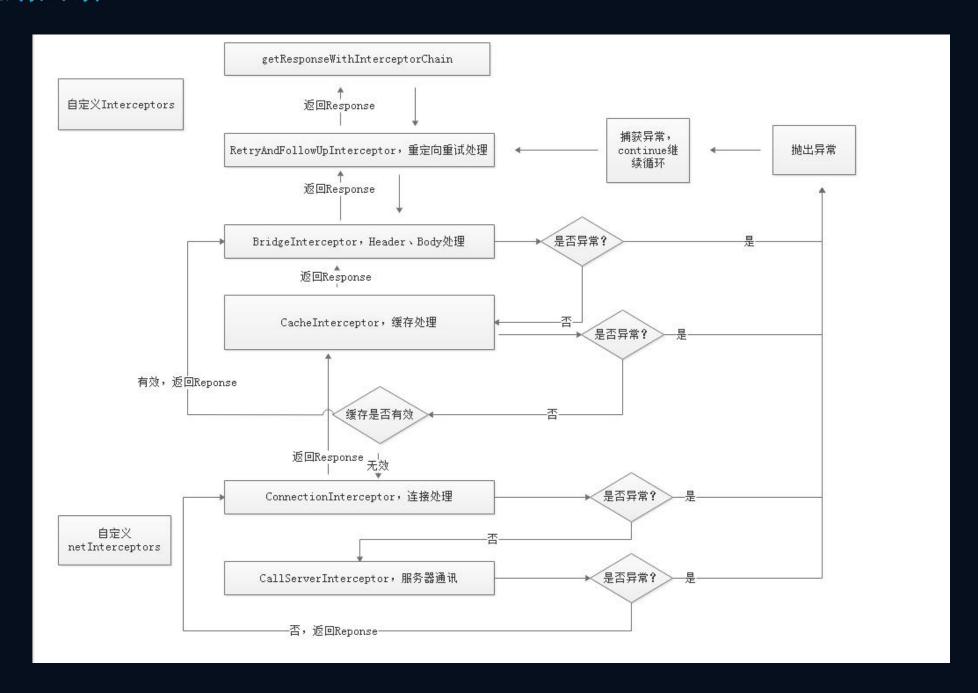
> Q: 如何决定将请求放入ready还是running?

➤ Q: 从ready移动running的条件是什么?

➤ Q: 分发器线程池的工作行为?

#### 上期回顾









重定向与重试拦截器

2

桥接拦截器

3

缓存拦截器

4

连接拦截器

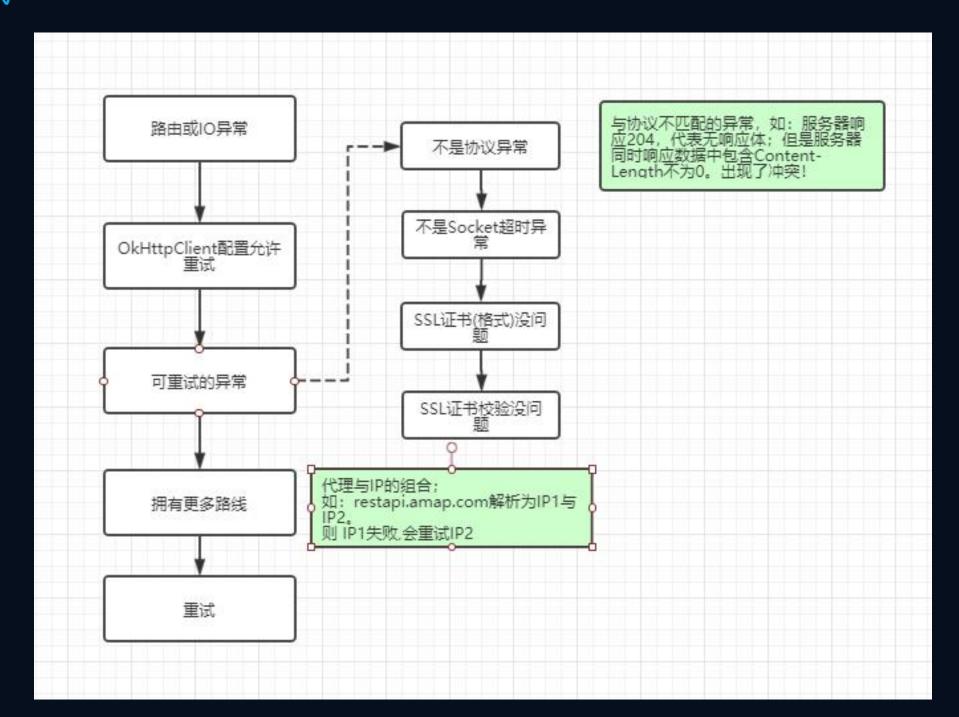
5

请求服务拦截器

享学课堂,让学习成为一种享受!



关注享学课堂 免费领取课程



#### ■重定向



响应码	说明	重定向条件
407	代理需要授权,如付费代理,需要验证 身份	通过proxyAuthenticator获得到了Request。例:添加 Proxy-Authorization 请求头
401	服务器需要授权,如某些接口需要登陆才能使用(不安全,基本上没用了)	通过authenticator获得到了Request。例:添加 Authorization 请求头
300、301、 302、303、 307、308	重定向响应	307与308必须为GET/HEAD请求再继续判断 1、用户允许自动重定向(默认允许) 2、能够获得 Location 响应头,并且值为有效url 3、如果重定向需要由HTTP到https间切换,需要允许(默认允许)
408	请求超时。服务器觉得你太慢了	1、用户允许自动重试(默认允许) 2、本次请求的结果不是响应408的重试结果 3、服务器未响应Retry-After(稍后重试),或者响应Retry-After: 0。
503	服务不可用	1、本次请求的结果不是响应503的重试结果 2、服务器明确响应 Retry-After: 0,立即重试





重定向与重试拦截器

2

桥接拦截器

3

缓存拦截器

4

连接拦截器

5

请求服务拦截器

享学课堂,让学习成为一种享受!



关注享学课堂 免费领取课程

#### ▮补全请求与响应后处理



#### 补全请求头

请求头	说明
Content-Type	请求体类型,如: application/x-www-form-urlencoded
Content-Length/Transfer-Encoding	请求体解析方式
Host	请求的主机站点
Connection: Keep-Alive	默认保持长连接
Accept-Encoding: gzip	接收响应体使用gzip压缩
Cookie	Cookie身份识别
User-Agent	用户信息,如:操作系统、浏览器等

#### 得到响应:

- 1、读取Set-Cookie响应头并调用接口告知用户,在下次请求则会读取对应的数据设置进入请求头,默认CookieJar无实现;
  - 2、响应头Content-Encoding为gzip,使用GzipSource包装解析。





重定向与重试拦截器

2

桥接拦截器

3

缓存拦截器

4

连接拦截器

5

请求服务拦截器

享学课堂,让学习成为一种享受!



关注享学课堂 免费领取课程

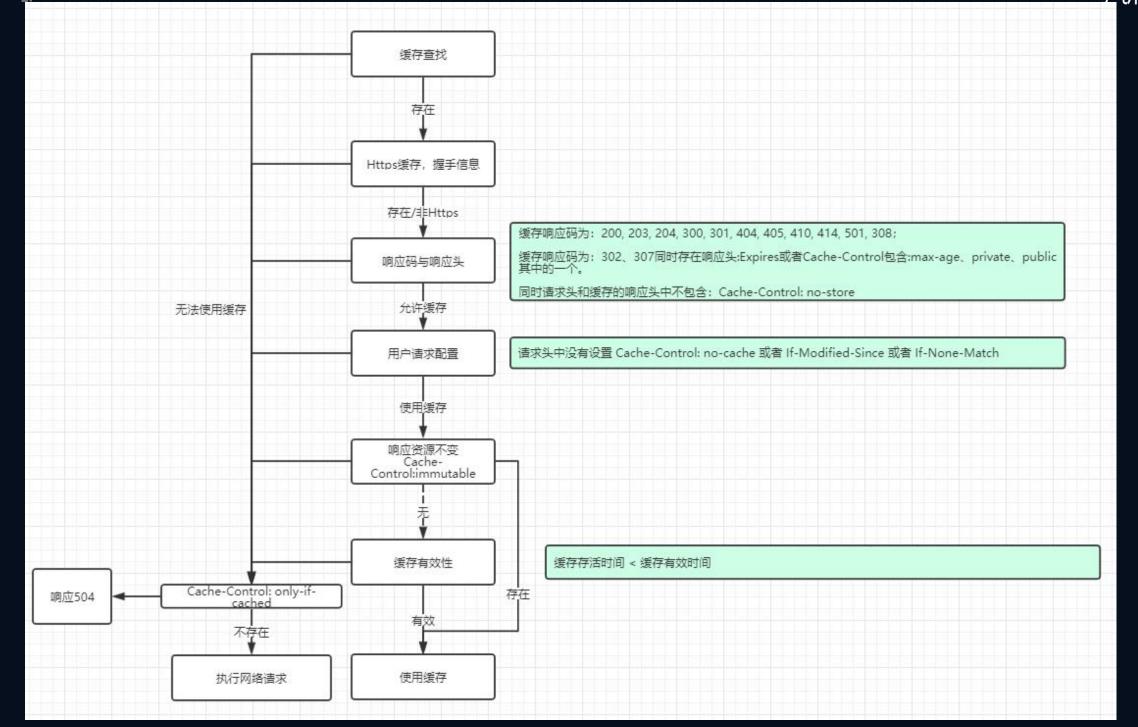
#### ▋缓存策略



拦截器通过CacheStrategy判断使用缓存或发起网络请求。此对象中的networkRequest与cacheResponse分别代表需要发起请求或者直接使用缓存

networkRequest	cacheResponse	说明
Null	Not Null	直接使用缓存
Not Null	Null	向服务器发起请求
Null	Null	直接gg,okhttp直接返回504
Not Null	Not Null	发起请求,若得到响应为304(无修改),则更新缓存响应并返回

即: networkRequest存在则优先发起网络请求,否则使用cacheResponse缓存,若都不存在则请求失败!







重定向与重试拦截器

2

桥接拦截器

3

缓存拦截器

4

连接拦截器

5

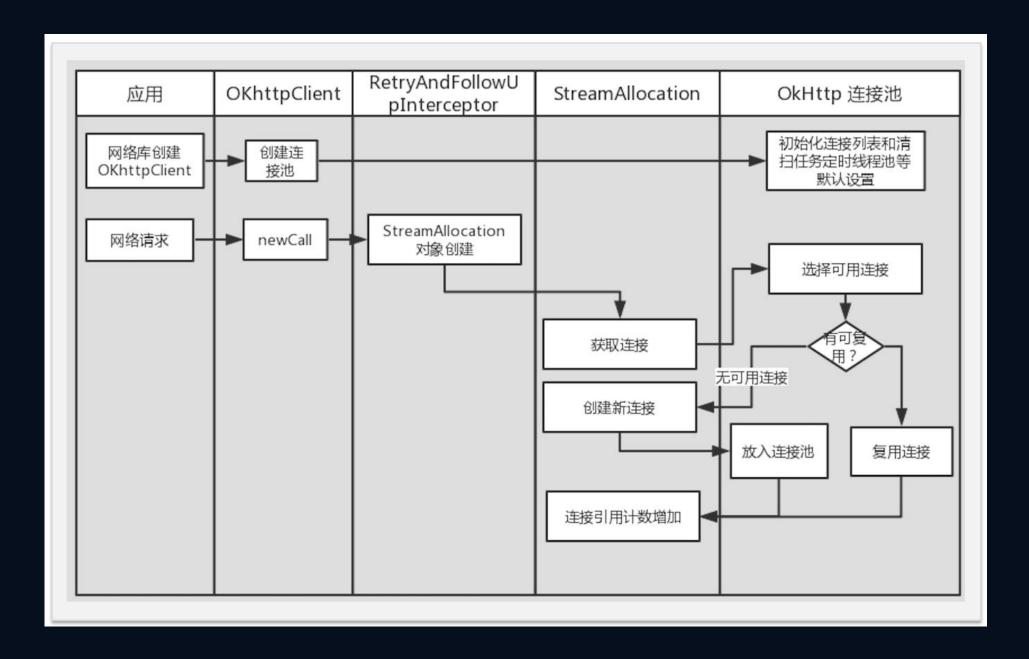
请求服务拦截器

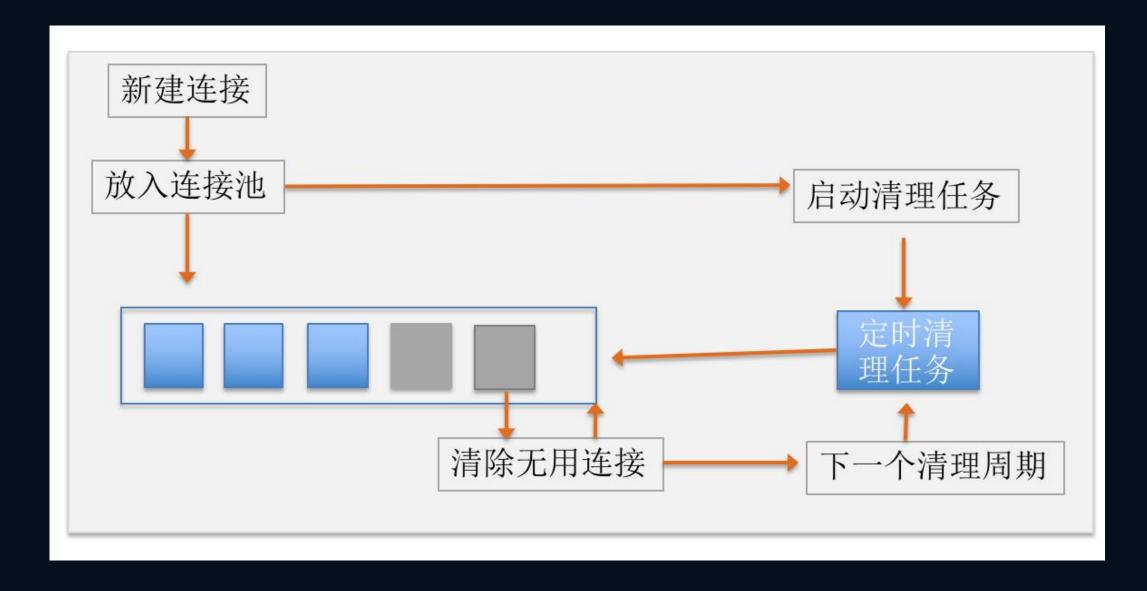
享学课堂,让学习成为一种享受!



关注享学课堂 免费领取课程









Socket socket = new Socket(); socket.connect( new InetSocketAddress("restapi.amap.com", 80)); 无代理 发送: GET /v3/weather/weatherInfo?city=长沙&key=13cb58f5884f9749287abbead9c658f2 HTTP/1.1 Proxy proxy = new Proxy(Proxy.Type.SOCKS,new InetSocketAddress("114.239.145.90", 808)) Socket socket = new Socket(proxy); SOCKS socket.connect(InetSocketAddress.createUnresolved("restapi.amap.com", 80)); 发送: GET /v3/weather/weatherInfo?city=长沙&key=13cb58f5884f9749287abbead9c658f2 HTTP/1.1 Socket socket = new Socket(); socket.connect(new InetSocketAddress("114.239.145.90", 808)); HTTP HTTP 发送: GET http://restapi.amap.com/v3/weather/weatherInfo?city=长沙&key=13cb58f5884f9749287abbead9c658f2 HTTP/1.1 Socket socket = new Socket(); socket.connect(new InetSocketAddress("114.239.145.90", 808)); HTTPS 发送: CONNECT restapi.amap.com HTTP/1.1





重定向与重试拦截器

2

桥接拦截器

3

缓存拦截器

4

连接拦截器

5

请求服务拦截器

享学课堂,让学习成为一种享受!



关注享学课堂 免费领取课程

#### Expect: 100-continue



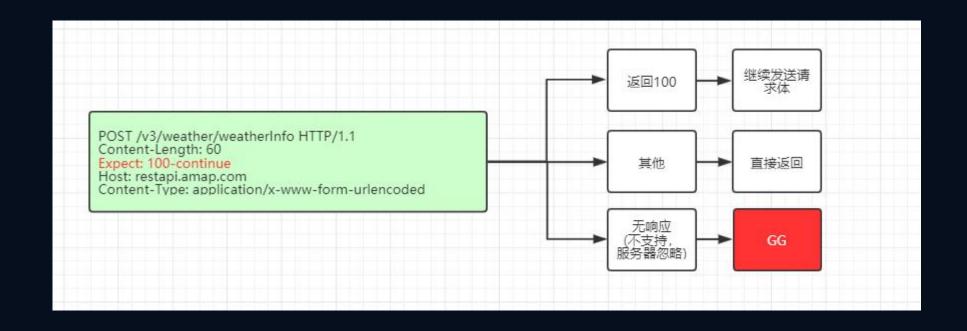


一般出现于上传大容量请求体或者需要验证。代表了先询问服务器是否原因接收发送请求体数据,

#### OkHttp的做法:

如果服务器允许则返回**100**,客户端继续发送请求体;如果服务器不允许则直接返回给用户。

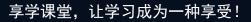
同时服务器也可能会忽略此请求头,一直无法读取应答,此时抛出超时异常。



# 谢娜观看

Liso

n



更多精彩课程: https://enjoy.ke.qq.com/



关注享学课堂 免费领取课程