http连接池相关参数设置：



* 1. cm.setDefaultSocketConfig(SocketConfig.custom().setSoKeepAlive(true).setSoTimeout(5000).build());

socket配置keepalive为true,超时时间为5s，和建立的长链接的过期时间保持一致。

* 1. cm.setMaxTotal(50);

cm.setDefaultMaxPerRoute(50);

配置最大连接为50，每个路由最大连接是也为50。

如果你要连接的url只有一个，两个必须配置成一样，否则只会取最小值。这也是为什么之前本地测试MaxTotal = 20 DefaultMaxPerRoute= 1时，用lsof命令看到本地与itemservice服务的连接数不是20而是1个的原因。

httpclient设置的最大连接数不能超过jetty设置的最大连接数，否则jetty的连接就会被httpclient连接池一直占用，直到系统挂掉，目前我们jetty容器最大连接数为256。

* 1. setKeepAliveStrategy(new ConnectionKeepAliveStrategy() {

…})

具体代码见上图。

配置keepalive策略，设置连接过期时间。

如果响应的response中有告诉我们多久过期，则使用其过期时间，如果没有指定，我们自己设置为5s过期，为了配合后面线程清空过期的连接。

1.4

HttpPost httpPost = *postForm*(**"http://"**+ host +**"/"**+url,json\_post);

httpPost.setHeader(**"Connection"**, **"Keep-Alive"**);

……

**finally** {  
 httpPost.releaseConnection();  
}

在post请求头中加入keepalive标识允许建立长连接。

httpclient使用完必须releaseconnection，不是abort。因为releaseconnection是归还连接到连接池，而abort是直接抛弃这个连接，而且占用连接池的数目。

1.5

setRetryHandler(new DefaultHttpRequestRetryHandler(3, true))

设置http client的重试次数是3次,也可以禁用掉。

1.6



 一般情况下服务器为了控制资源开销，不会永久的维护一个长连接，过一段时间就会关闭该连接。如果放回连接池的连接，如果在服务器端已经关闭，客户端是无法检测到这个状态变化而及时的关闭Socket的，造成了线程从连接池中获取的连接不一定是有效的。PoolingHttpClientConnectionManager提供了closeExpiredConnections()方法和closeIdleConnections()方法来解决相应的问题。在我们代码中，我们会起一个后台线程每隔5s执行一次，去关闭线程池中过期以及空闲超过30s的连接。通过HttpClient的setKeepAliveStrategy方法设置连接的过期时间，配合closeExpiredConnections()方法解决连接池中连接失效的。