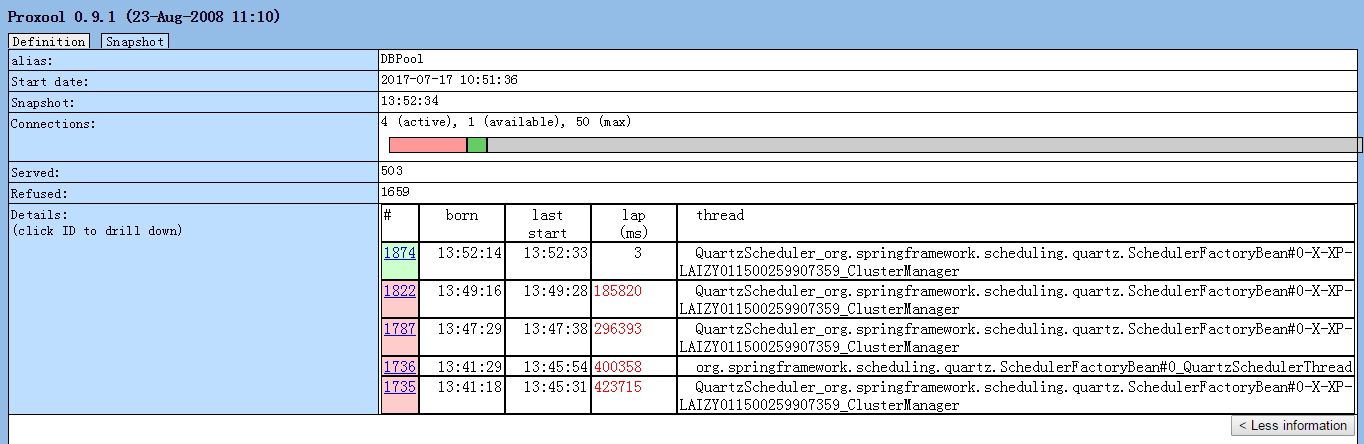
# ProxoolDataSource数据库连接池

管理connection的一个大概类图



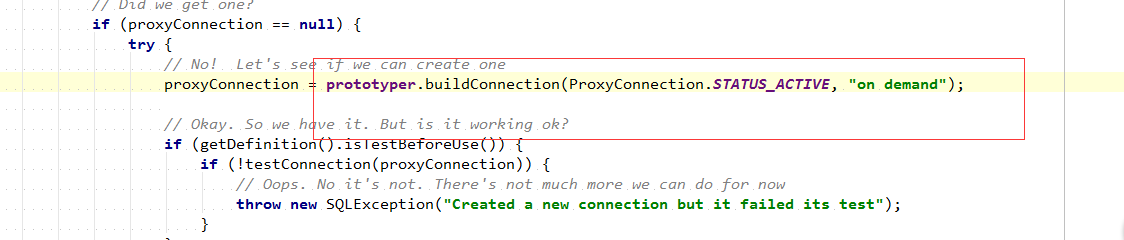
问题：proxool中的连接一直处于活跃状态，被占用无法释放，如下图



问题原因：

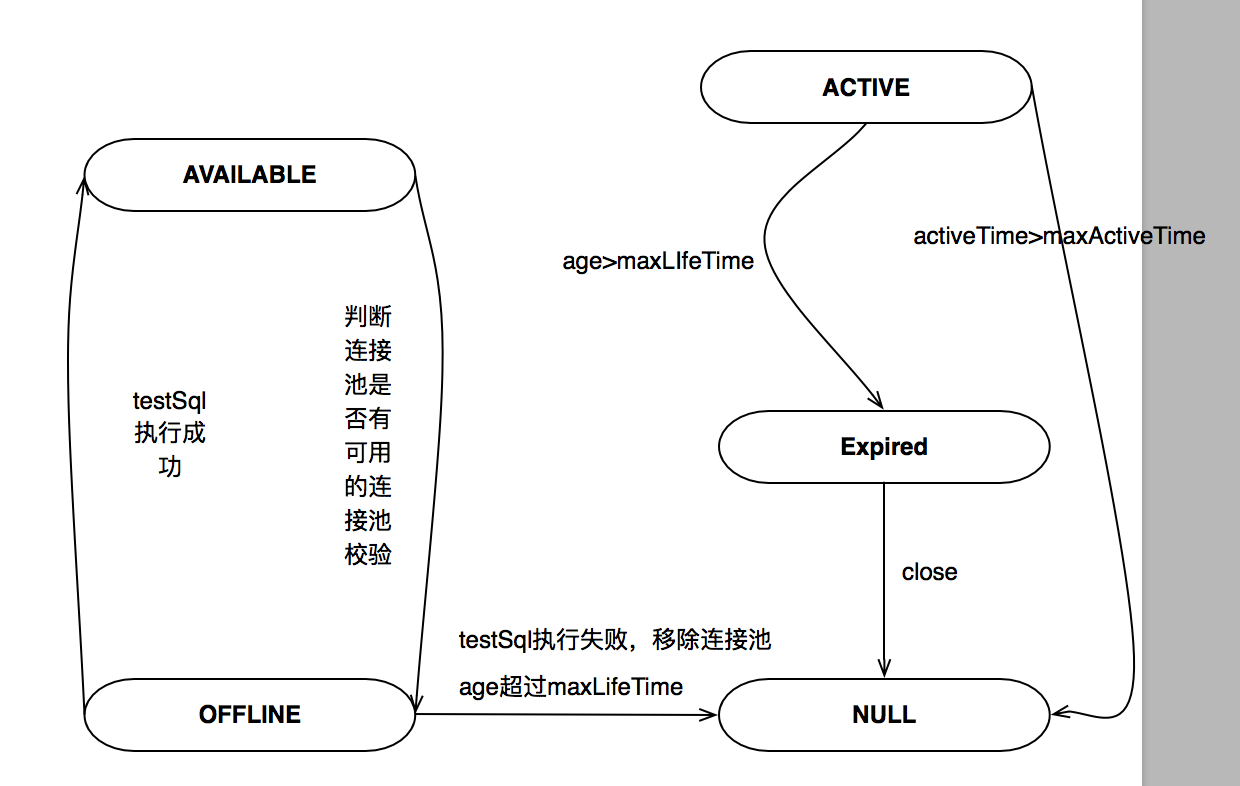


由于只配了最小连接，没有配置可用的连接配置。导致无法在PrototyperThread后台线程在创建的时候，不会去创建连接，而是等到需要getConnection的时候才会buildConnection



ConnectionPool

## 状态机



# Auto Vacuum

autovacuum\_analyze\_scale\_factor=0.1

autovacuum\_analyse\_threshold=50

以上两个配置：比如当前表记录有10000条，那么需要触发10000\*0.1+50=1050条的增删改才会触发autovacuum\_analyse;

查询autovacuum\_analyse的记录

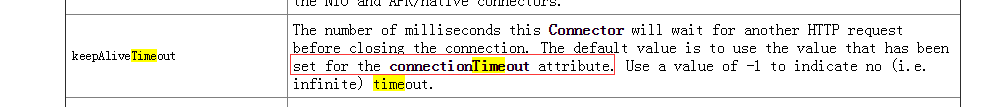
select \* from pg\_stat\_user\_tables;

如果新增字段的话，会不会触发autovacuum呢？

不会；应该是新增字段，是不会触发新增、删除、或者修改

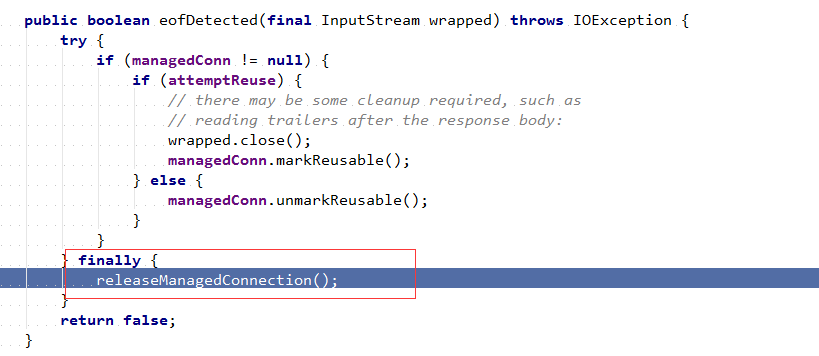
# HttpClient连接池

httpClient默认是保持长连接的。而tomcat中默认的最大长连接时间是connectionTimeout，如图tomcat的文档说明



## HttpClient自动releaseConnection

如果有去解析httpResponse的话，会自动释放连接



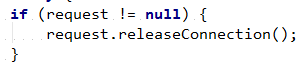
BasicManagedEntity

但是这里的释放并不会去关闭socket连接，只会将connection重新放回到连接池中。这样会导致后来手动调用releaseConnection会无效。代码原因如下：



ManagedClientConnectionImpl

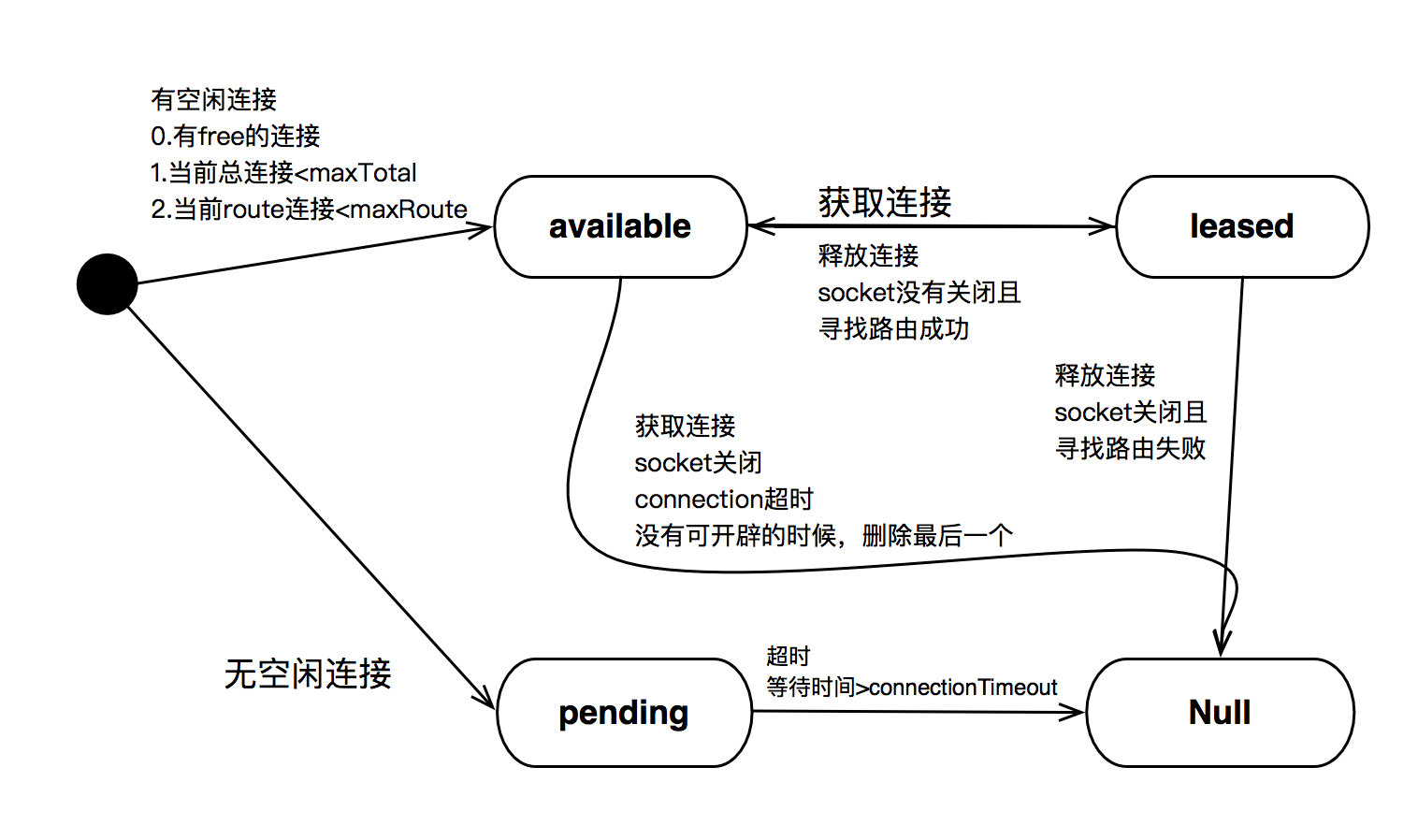
## 手动去执行releaseConnection



会去close socket。

pool的机制是：每次取connection的时候，才去维护connection的超时。并没有一个后台线程会实时去扫

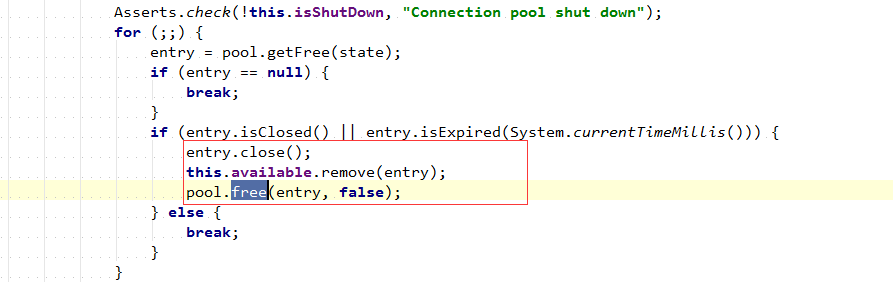
## 状态机



AbstractConnPool.getPoolEntryBlocking

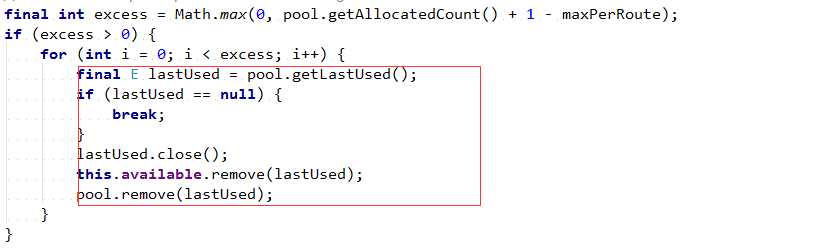
### available->null: 有三种情况；

1. 当前连接被关闭或者已超时，如下图



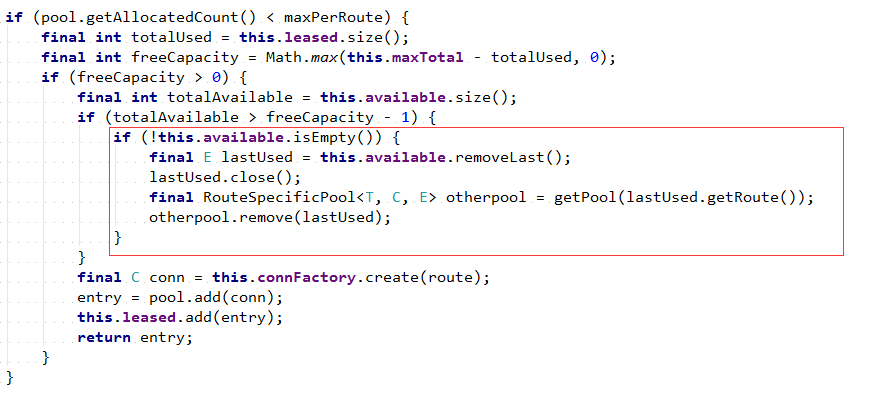
AbstractConnPool.getPoolEntryBlocking代码片段

1. 当分配的时候超出了本池的可开辟的空间，需释放旧的available以释放空间(每一个route都可以由http.user-token)区分开。所以会有available但却获取不到



AbstractConnPool.getPoolEntryBlocking代码片段

1. 当前所有route的available已经超出最大个数，而当前route需要开辟一个新的available。所以需要释放旧的available状态



AbstractConnPool.getPoolEntryBlocking代码片段

### leased->null

没有处理response，httpRequest.releaseConnection，由用户主动close的时候，不会被重新释放到连接池，而是变成null，如图0

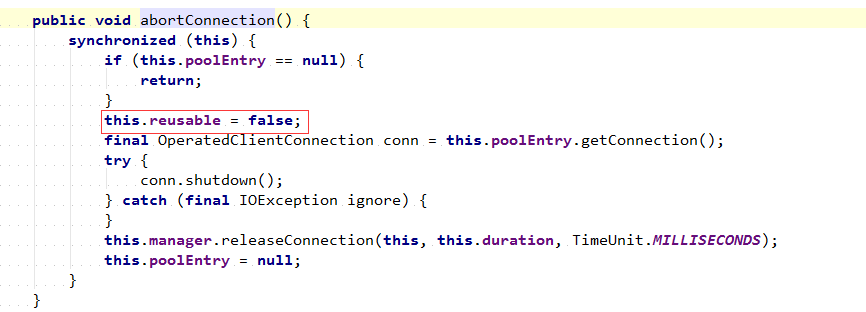


图0 ManagedClientConnectionImpl的代码

### leased->available

处理了response，会由response中关闭流(BasicManagedEntity. eofDetected)的，leased在释放的时候，会被重新放入池中，从而变成availabel状态，等待获取。由服务器下发的Keep-Alive的cookie信息来保存过期时间(如图1)使用

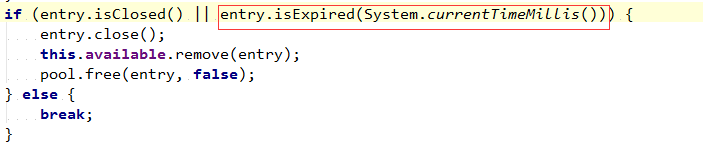


图1 AbstractConnPool.getPoolEntryBlocking代码片段