## RaftClient

### Class ClientSocket {

* public Socket socket;
* public void write(String msg) throws IOException {} 发送消息
* public String read() throws IOException {} 接收消息
* public void close() {} 关闭socket

### }

### Class Client {

* + private ClientSocket clientSocket 和服务器连接的socket
  + private List<String> serverAddrList 服务器地址列表
  + private serverNum 服务器数量
  + private tryTimes 尝试次数 10
* private boolean tryConnectingLeader { 尝试与leader建立连接

for(遍历每一个server的地址) {

建立连接，生成clientSocket

发送client连接消息

调用clientSocket.read, 读取结果

if(返回连接成功) {

return true;

}

}

return false;

}

* public void sendCommand(String command, boolean read, Callback callback) {

int count = 0;

String commandId 生成这一条command对应的commandId

while(count < tryTimes) {

++count

if(tryConnectingLeader) {

打包msg8

try {

clientSocket.write() 向服务器发送指令

} catch(IOException e) {

关闭clientSocket

continue

}

try {

clientSocket.read() 读取服务器响应

if(是正常的指令响应) {

执行之后的逻辑

If( read ) { 是数据库查询指令

callback.callback(返回结果)

}

} else { 是由于leader降级带来的消息

continue

}

} catch(IOException e) {

关闭clientSocket

continue

}

} else {

输出 服务器不稳定

break

}

}

}

### }

### Class CallBack {

* public void callback(List<Map<String, Object>> results) {} 处理返回的结果

### }