

- 1、发送心跳消息
- 2、接收心跳响应消息

```
1 //ConnClientChannelHandler.java
2
3 @Override
4 public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {
5     connection.updateLastReadTime();
6     if (msg instanceof Packet) {
7         Packet packet = (Packet) msg;
8         Command command = Command.toCMD(packet.cmd);
9         if (command == Command.HANDSHAKE) {
10             //发送心跳消息
11             startHeartBeat(message.heartbeat - 1000);
12         } else if (command == Command.FAST_CONNECT) {
13             //发送心跳消息
14             startHeartBeat(message.heartbeat - 1000);
15         } else if (command == Command.KICK) {
16             ...
17         } else if (command == Command.ERROR) {
18             ...
19         } else if (command == Command.PUSH) {
20             ...
21         } else if (command == Command.HEARTBEAT) {
22             //接收心跳响应消息
23             LOGGER.info("receive heartbeat pong...");
24         } else if (command == Command.OK) {
25             ...
26         } else if (command == Command.HTTP_PROXY) {
27             ...
28         }
29     }
30     LOGGER.debug("receive package={}, chanel={}", msg, ctx.channel());
31 }
32
```

发送心跳消息：

```
1 //ConnClientChannelHandler.java
```

```

2
3 private static final Timer HASHED_WHEEL_TIMER = new HashedWheelTimer(new
NamedPoolThreadFactory(ThreadNames.T_CONN_TIMER));
4 //创建心跳任务
5 private void startHeartBeat(final int heartbeat) throws Exception {
6     HASHED_WHEEL_TIMER.newTimeout(new TimerTask() {
7         @Override
8         public void run(Timeout timeout) throws Exception {
9             //如果是连接状态，且健康检查成功，则继续心跳检测
10            if (connection.isConnected() && healthCheck()) {
11                HASHED_WHEEL_TIMER.newTimeout(this, heartbeat, TimeUnit.MILLISECONDS);
12            }
13        }
14    }, heartbeat, TimeUnit.MILLISECONDS);
15 }
16 //健康检查
17 //如果可读、可写，则是健康的；
18 //如果不可读，且超过2次计数，则断开连接；
19 //如果不可写，发送心跳包；
20 private int hbTimeoutTimes; //心跳超时次数
21 private boolean healthCheck() {
22     //如果读取超时，累计超时次数
23     if (connection.isReadTimeout()) {
24         hbTimeoutTimes++;
25         LOGGER.warn("heartbeat timeout times={}, client={}", hbTimeoutTimes, co
nnection);
26     } else {
27         hbTimeoutTimes = 0;
28     }
29     //心跳超时次数超过2次，断开连接，返回健康检查失败
30     if (hbTimeoutTimes >= 2) {
31         LOGGER.warn("heartbeat timeout times={} over limit={}, client={}", hbTi
meoutTimes, 2, connection);
32         hbTimeoutTimes = 0;
33         connection.close();
34         return false;
35     }
36     //如果写超时(有可能网络不稳定或者网络断开)，发送心跳包
37     if (connection.isWriteTimeout()) {
38         LOGGER.info("send heartbeat ping...");
39         connection.send(Packet.HB_PACKET); //发送心跳包

```

```
40  }  
41  return true;  
42  }
```